all rights reversed

HACKER

Φλεβάρης 2014 # Τιμή τεύχους 4€

τεύχος 029 - Aurora Invasion Edition

pfSense

Φανταστικός router με load balancing [*]

SQL injection για αρχάριους

Δώστε διαδικτυακή φωνή στο παλιό τρανζιστοράκι

Στήστε τον δικό σας time server, για τον κόσμο όλο

[*] και θα θέλετε κι εσείς 2 γραμμές Internet στο σπίτι!

all rights reversed

HACKER

tefhos 29o

Περιεχόμενα

και οδηγίες χρήσης

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Πέτρος Κυλαδίτης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το ηλεκτρονικό περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55133, Καλαμαριά. Ο δικτυακός τόπος του περιοδικού είναι στο http://deltahacker.gr. Το email επικοινωνίας είναι το talk2us@deltahacker.gr και ειδικά για τις συνδρομές είναι το subscriptions@deltahacker.gr. Οι γενικές πληροφορίες για τις συνδρομές παρατίθενται στο http://deltahacker.gr/subscriptions και οι ίδιες οι παραγγελίες γίνονται από το http://deltahacker.gr/pdf12order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) Τι λέτε; Φτιάχνουμε κόμμα;



delta HACKER
Στη 3η χρονιά και συνεχίζουμε ακάθεκτοι!
Μηνιαίο περιοδικό deltaHacker, http://deltahacker.gr
Ethical Hacking / Δίκτυα / Ασφάλεια / Προγραμματισμός / Ηλεκτρονικά
twitter.com/deltaHacker
facebook.com/deltaHacker

Το μυστικό του Θείου και μια εξόρμηση από το Βορρά

Αγαπητοί μου,

Αυτό που κάνω τώρα, από καιρό το σχεδίαζα.

Κατ' αρχάς οφείλω να σας πω ότι δεν γράφω το editorial τούτο από τα κεντρικά της Parabing Creations, στο Northern Outpost, στη Θεσσαλονίκη.

Ούτε από το Πόρτο Ράφτη, το γράφω.

Από χθες βρίσκομαι λίγο βορειότερα, για την ακρίβεια στη Δράμα, όπου έχουμε έρθει για μερικές μέρες με τη Σύμβουλο Εκδόσεως (TM) του περιοδικού, aka @citr0nella.

Η Δράμα έχει πολλά καλά κι ένα απ' αυτά είναι ότι ο χειμώνας, που πολύ μ' αρέσει και τον προτιμώ από το καλοκαίρι, κρατάει λίγο περισσότερο απ' όσο κρατάει, π.χ., στη Θεσσαλονίκη.

Κατά δεύτερον, να σας πω ότι γι' αυτό και μόνο το editorial προσπαθώ να γράψω όπως γράφει ο Θείος Ακάκιος.

Αυτό ακριβώς είναι που από καιρό σχεδίαζα, όχι το να έλθουμε στη Δράμα με τη Σύμβουλο Εκδόσεως (TM).

Ούτως ή άλλως η Δράμα είναι κοντά στη Θεσσαλονίκη, οπότε συχνά ερχόμαστε και ιδιαίτερος σχεδιασμός δεν χρειάζεται.

Εκτός ίσως από το να πάρεις τηλέφωνο μια μέρα πριν στο ΚΤΕΛ "Μακεδονία", για να κλείσεις θέσεις.

Δεν θέλετε να πάτε στο ΚΤΕΛ "Μακεδονία" τελευταία στιγμή, χωρίς να έχετε κλείσει θέσεις. Δηλαδή αν θέλετε μπορείτε ελεύθερα να το κάνετε, ωστόσο καλό είναι να το αποφεύγετε.

Αλλά ας μην ξεφεύγω.

Κάποιο μυστικό έχει ο Θείος Ακάκιος, που λέτε, και μέσω των επιστολών που μάς στέλνει και δημοσιεύουμε συχνά ξεσηκώνει αντιδράσεις.

Δεν είμαι σίγουρος αν γνωρίζω το μυστικό του. Φορές τείνω να πιστεύω ότι έχει να κάνει και με τις συχνές αλλαγές παραγράφων, σαν αυτές που κι εγώ, τώρα, εντελώς ερασιτεχνικά προσπαθώ να κάνω.

Υποψιάζομαι βέβαια ότι δεν είναι μόνο οι συχνές αλλαγές παραγράφων.

Πρέπει να 'ναι και η κρυφή τέχνη του *πότε* ν' αλλάζεις παράγραφο.

Να, κάπως έτσι.

Είδατε τι έκανα;

Χμ, ναι, έχετε δίκιο.

Μάλλον δεν το 'χω.

Αλλά αφού το ξεκίνησα έτσι, ας προσπαθήσω να το τελειώσω κι από το επόμενο editorial επιστρέφω στο παλιό, γνώριμο κι ασφαλές για 'μένα στιλ. Είναι και που, σε αντίθεση με τον Θείο, δεν χρησιμοποιώ γραφομηχανή.

Αυτή τη στιγμή, ας πούμε, πληκτρολογώ τις γραμμές αυτές σε ένα MacBook 6ετίας.

"Σιγά τη διαφορά που έχει από γραφομηχανή", θα μου πείτε.

Αν πράγματι μου το πείτε, δίκιο θα έχετε.

(Λέτε τελικά να το 'χω κι εγώ με τις αλλαγές παραγράφων;)

Όση ώρα γράφω, που λέτε, έχω ανοίξει κι ένα SSH tunnel προς τον router που εδώ και λίγες μέρες έχουμε στήσει στο Northern Outpost. Πατάω συχνά, λοιπόν, [Command+Tab] και ρίχνω κλεφτές ματιές στο wep panel του pfSense -- και χαίρομαι.

Αυτό είναι το λειτουργικό του νέου μας router, αγαπητοί μου. Ένα από τα πολλά του "συν" είναι το load balancing. Στην ουσία, το box χρησιμοποιεί δύο γραμμές προς το Internet.

Μεγάλη πώρωση κι όλες τις λεπτομέρειες του σχετικού πρότζεκτ θα τις βρείτε στο τεύχος 029 του deltaHacker, το editorial του οποίου τώρα διαβάζετε.

Βέβαια παίζοντας με τον ολοκαίνουργιο router και κάνοντας διάφορες δοκιμές, ώστε να διαπιστώνω τη σημαντική βελτίωση στις ταχύτητες download/upload, λησμόνησα να στήσω στο pfSense το OpenVPN.

Γι' αυτό και τώρα έχω αναγκαστεί να καταφύγω στο SSH tunneling, το επονομαζόμενο "VPN του φτωχού".

Δεν ξέρω αν όντως είναι "του φτωχού", πρέπει όμως να ομολογήσω ότι μια χαρά δουλεύει. Αλλά εντάξει, θέλω οπωσδήποτε να στήσω το OpenVPN στο pfSense -- και φυσικά θα γράψω και σχετικό αρθράκι.

Εκτός κι αν αντί για άρθρο ετοιμάσω επεισόδιο για το deltaCast. Χμ. Καλύτερη ιδέα μου φαίνεται αυτή. Θα βάλω κι άλλα στο επεισόδιο, όπως, π.χ., ρύθμιση του Squid για να λειτουργεί με το load balancing.

Άντε δηλαδή μη βάλω και το Snort, στο ίδιο επεισόδιο! Θα βγει βέβαια λίγο μεγάλο, αλλά αυτό δεν είναι κάτι νέο για το deltaCast.

Τι τα θέλετε, βρήκα πάλι λόγους για να ενθουσιαστώ, με αφορμή το νέο τεύχος.

Πρέπει βέβαια να ομολογήσω ότι είναι κι άλλα που με κρατάνε ενθουσιασμένο, τον τελευταίο καιρό.

Να, εντελώς στην τύχη, θα αναφέρω την αγγλική έκδοση του περιοδικού, την οποία ήδη έχουμε στα σκαριά.

Ο κύριος Αρχισυντάκτης (TM), aka @pvar, μαζί με τον DTP Guy (TM), aka @ petrarakos, έχουν αναλάβει τη σχεδίαση του νέου site.

Αποφάσισαν να ξεκινήσουν από μηδενική, πρακτικά, βάση. Το site δεν είναι ακόμα στον αέρα --μην το ψάξετε--, όμως μπορώ να σας διαβεβαιώσω ότι θα κάνει εντύπωση.

Θετική.

Το άλλο με την αγγλική έκδοση, σας το είπα; Κάτι πρέπει να 'χω αναφέρει σε Twitter και Facebook, αλλά αξίζει --και θέλω-- να το επαναλάβω.

Ως νεοσύστατη εταιρεία, που λέτε, γίναμε δεκτοί στο Truly Green (TM) Startup Program της GreenQloud, οι άνθρωποι της οποίας ανέλαβαν να μας παράσχουν την υποδομή που χρειαζόμαστε για το νέο περιοδικό. Τις προάλλες είχαμε κανονίσει και ραντεβού, μέσω Skype, όπου τα είπαμε και μας υποδέχθηκαν επισήμως στο πρόγραμμα.

Ίσως εδώ πρέπει ν' αναφέρω δύο λεπτομέρειες που, για 'μένα τουλάχιστον, κάνουν τη διαφορά.

Λεπτομέρεια 1η: Η GreenQloud είναι στην Ισλανδία. Από εκεί θα ξεκινάει η αγγλική έκδοση του deltaHacker -- και θ' απευθύνεται σ' ολόκληρο τον πλανήτη.

Μιλάμε για δυναμική εξόρμηση, αλλά και καθ' όλα φιλική. Φιλική προς τους αναγνώστες, φιλική κι ως προς το περιβάλλον.

Λεπτομέρεια 2η: Φιλική ως προς το περιβάλλον διότι, στη GreenQloud, χρησιμοποιούν αποκλειστικά ανανεώσιμες και με μηδενικό αντίκτυπο στο περιβάλλον μορφές ενέργειας (γεωθερμική και υδροηλεκτρική).

Δεν πρόκειται για μαρκετίστικο κόλπο. Έτσι ακριβώς έχουν τα πράγματα.

Όσοι διαβάζετε, έστω και περιστασιακά, το editorial, θα 'χετε παρατηρήσει ότι είμαι οπαδός του Βορρά, του χειμώνα και της Ισλανδίας. Κάτι που έχω αποφύγει ν' αναφέρω είναι ότι είμαι κι ένθερμος υποστηρικτής των περιβαλλοντικά φιλικών μορφών ενέργειας.

Είναι λοιπόν να μην ενθουσιάζομαι, τον τελευταίο καιρό;

Άσχετο, αλλά συνειδητοποιώ τώρα ότι αυτές οι συχνές αλλαγές παραγράφων βοηθούν σημαντικά.

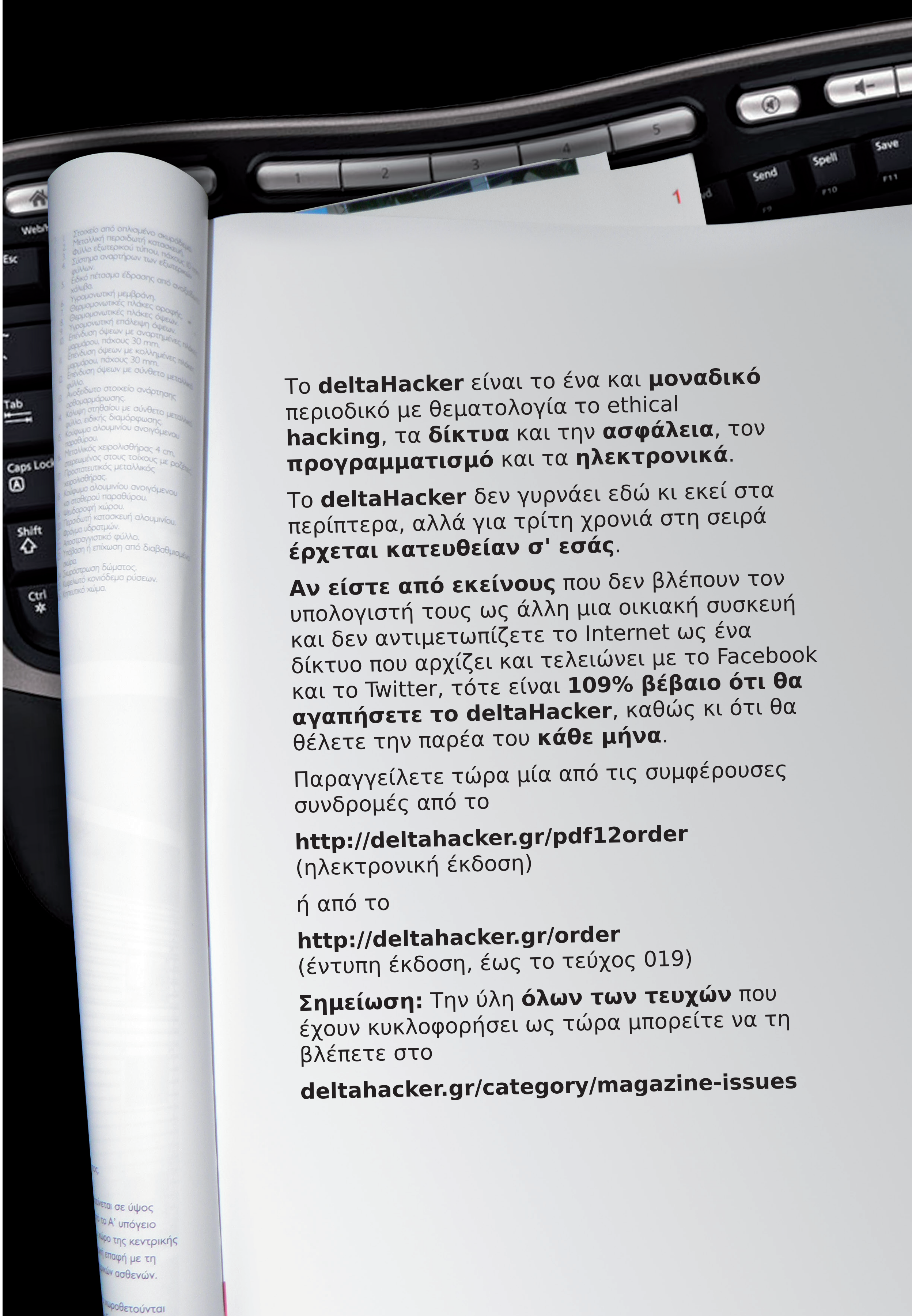
Αν μη τι άλλο, βάζεις τις σκέψεις σου πιο εύκολα σε μια σειρά.

Όχι ότι ο Θείος Ακάκιος έχει ανάγκη να βάζει σε σειρά τις σκέψεις του, δηλαδή.

Σε αντίθεση με κάποιους από εμάς.

Σας ασπάζομαι,

Ο Προκομμένος Σας (TM)


Το deltaHacker είναι το ένα και μοναδικό περιοδικό με θεματολογία το ethical hacking, τα δίκτυα και την ασφάλεια, τον προγραμματισμό και τα ηλεκτρονικά.
Το deltaHacker δεν γυρνάει εδώ κι εκεί στα περίπτερα, αλλά για τρίτη χρονιά στη σειρά έρχεται κατευθείαν σ' εσάς.
Αν είστε από εκείνους που δεν βλέπουν τον υπολογιστή τους ως άλλη μια οικιακή συσκευή και δεν αντιμετωπίζετε το Internet ως ένα δίκτυο που αρχίζει και τελειώνει με το Facebook και το Twitter, τότε είναι 109% βέβαιο ότι θα αγαπήσετε το deltaHacker, καθώς κι ότι θα θέλετε την παρέα του κάθε μήνα.
Παραγγείλετε τώρα μία από τις συμφέρουσες συνδρομές από το
http://deltahacker.gr/pdf12order
(ηλεκτρονική έκδοση)
ή από το
http://deltahacker.gr/order
(έντυπη έκδοση, έως το τεύχος 019)
Σημείωση: Την ύλη όλων των τευχών που έχουν κυκλοφορήσει ως τώρα μπορείτε να τη βλέπετε στο
deltahacker.gr/category/magazine-issues



σνέιλ μέιλ

Αγαπητοί μου,

Σας γράφω βραδάκι (τσικνο)Πέμπτης καθισμένος στο μεγάλο τραπέζι στο σαλόνι. Η πόρτα προς το μπαλκόνι πίσω μου είναι ανοιχτή.

Ο καιρός είναι ανοιξιάτικος και από παντού μας πλημμυρίζει η τσίκνα.

Στην κουζίνα επικρατεί μια αναστάτωση. Η πόρτα είναι κλειστή για το κοινό, και οι επιφορτισμένες με τα της μαγειρικής κυρίες της οικίας μας έχουν αποφασίσει να συνδυάσουν τις παραδόσεις της Τσικνοπέμπτης με τις τελετουργίες του Σίγκμο Χόλι, της ανοιξιάτικης γιορτής των χρωμάτων από την Γκόα.

Δεν ξέρω τι έχουν κατά νου αλλά από το πρωί ετοίμαζαν νάαν, κρέας για μαρινάρισμα, τσίλι με μπαχαρικά και πιπερόριζα, και ρύζι. Έχουν φέρει και πιατάκια με χρωματιστές σκόνες (κίτρινη κουρκουμίνη, μπλε ιβίσκο, πορφυρό αποξηραμένο πατζάρι, καφέ τριμμένα φύλλα τσαγιού και κάρβουνο) κι από ό,τι έχω καταλάβει ετοιμάζονται, κατά το έθιμο, να μας πασαλείψουν την ώρα που εμείς θα τσικνίζουμε αμέριμνοι. Αυτό μένει να το δούμε.

Γενικώς αυτές τις μέρες ανυπομονούσα να σας γράψω. Για να καταλάβετε, είχα προγραμματίσει να ξεκινήσω την επιστολή μου τις προάλλες, ανήμερα του Αγίου Βαλεντίνου.

Όχι, δεν είχα στο μυαλό μου να γράψω περί του ξενόφερτου Αγίου και λοιπά και λοιπά. Τέτοια θεματολογία δεν απασχολεί ούτε τους αναγνώστες, ούτε τους συντάκτες, εικάζω, του περιοδικού αυτού.

Ακολουθώντας το νήμα από εκεί που το είχα αφήσει την περασμένη φορά, θα σας έγραφα για τον Όκκαμ και το ξυράφι του.

Πάω, που λέτε, να πιάσω μια λευκή σελίδα για να τη βάλω στην γραφομηχανή, και βλέπω μιαν εκτύπωση από το ίντερνετ, πιασμένη με συνδετήρα, και κολλημένο πάνω της ένα χαρτάκι post-it. Ο γραφικός χαρακτήρας ήταν της εξαδέλφης.

«Για ρίξε μια ματιά» με παρότρυνε το χαρτάκι.

Ρίχνω μια ματιά, και τι να δω; Η επιστολή μου. Η πιο πρόσφατη, αυτή με τον Όκκαμ. Εκτυπωμένη από το ίντερνετ.

Μου το είχε πει βέβαια ο Κύριος Εκδότης, ότι σκεφτόταν να περάσει όλο το αρχείο των επιστολών στο διαδίκτυο, αλλά δεν είχα συνειδητοποιήσει ότι το έλεγε στα σοβαρά.

«Τι λε ρε παιδί μου, βρίσκεις τέτοια πράγματα στο ίντερνετ;» αναρωτήθηκα, φωναχτά σχεδόν, κι έβαλα στην άκρη τη γραφομηχανή.

Θυμήθηκα το θαυμασμό και το θάμπωμα που ένοιωθα τα παλιά εκείνα χρόνια, τότε που έμπαινες στο δίκτυο με μόντεμ των 1200bps και έκανες βόλτες με gopher και lynx. Με έπιασε η νοσταλγία.

Εκείνες ήταν εποχές, αγαπητοί μου.

Θυμάμαι, λες και ήταν χτες, τις διαφωνίες και τους καυγάδες με τον φίλο μου τον Δ. (TM). Εγώ συμπαθούσα το gopher. Αυτουνού του άρεσε το lynx.

Το gopher είχε μια ιεραρχικότητα: Μενού, υπομενού, καταλόγους, υποκαταλόγους, λίστες -- συνδεόσουν και εντόπιζες αμέσως την πληροφορία που ήθελες. Εύκολα. Απλά. Συγκεκριμένα.

Το lynx δεν το χώνευα και τόσο. Όλες αυτές οι σελίδες με τα χρωματιστά και τα έντονα γραμματάκια έδειχναν έλλειψη σοβαρότητας, τουλάχιστον έτσι μου φαινόταν εμένα, και τις περιφρονούσα. Ο Δ., ωστόσο, επέμενε πως εκεί βρίσκεται το μέλλον.

Εκεί που τα βρίσκαμε με τον Δ. ήταν στην αγάπη μας για το ftp. Το αγαπούσαμε. Ηταν το βασικό μας εργαλείο.

Εγώ δούλευα σε Windows 3.1 τότε και χρησιμοποιούσα το ProComm Plus -- ένα πρόγραμμα προσομοίωσης κονσόλας τερματικού που σου έδινε telnet, ftp και όλα τα καλά.

Τα πράγματα άρχισαν να αλλάζουν όταν εμφανίστηκε το Mosaic.

Το Mosaic χρειαζόταν να υποστηρίζεται το TCP/IP. Τέτοιες δυνατότητες τα Windows 3.1 δεν είχαν. Επρεπε να εγκαταστήσεις πρώτα τις βιβλιοθήκες winsock. Αυτό γινόταν εύκολα, αλλά σπανίως γινόταν σωστά. Απλά, τα Windows 3.1 δεν ήταν ακόμα ώριμα για δικτύωση, και μάλιστα για δικτύωση με πρωτόκολλο TCP/IP. Ταλαιπωρία.

Τέλος πάντων, για να μην τα πολυλογώ, με τα πολλά τα καταφέρναμε και το Mosaic ξεκινούσε.

Το Mosaic αποτέλεσε αυτό που λένε paradigm shift. Αλλαγή παραδείγματος. Από τη στιγμή που μπήκε στο παιχνίδι, το παιχνίδι άλλαξε. Οι εξελίξεις ακολούθησαν τον δικό τους δρόμο, έναν δρόμο καινούργιο που άνοιγε έτσι όπως προχωρούσαμε: Mosaic, Netscape, Firefox

The rest is history, που λένε και οι φίλοι μας οι ξένοι.

Αυτά σκεφτόμουν ανήμερα του Αγίου Βαλεντίνου, αγαπητοί μου, και τελικά επιστολή δεν σας έγραψα.

Σήμερα, κοτζάμ Τσικοπέμπτη, παραμένω στο ίδιο εκείνο mood, της εποχής του πρώιμου ιντερνετ, μιας εποχής γεμάτης ελπίδα και αισιοδοξία.

Αφού με βλέπει χαμογελαστό η μητέρα της εξαδέλφης, που είναι ινδουίστρια, και έχει μπερδευτεί.

Το πρώτο μου μόντεμ αντάλλασσε δεδομένα στα 1200bps. Μου το είχε δώσει κάποιος φίλος όταν αναβάθμισε το δικό του και ανέβηκε στα 2400bps. Ακόμα και στα 1200bps, όμως, ο ενθουσιασμός που νοιώθαμε όταν καταφέρναμε με τα πολλά να συνδεθούμε με κάποιον άλλο υπολογιστή ήταν κάτι που αν δεν το έχεις ζήσει δεν μπορείς να το καταλάβεις.

Το μόνο με το οποίο μπορώ εγώ να συγκρίνω την αίσθηση εκείνη είναι το ραδιόφωνο στα βραχέα κύματα, και η χαρά που ένοιωθα όταν κάποια βράδια με καλό καιρό κατάφερνα να πιάσω εκπομπές από κάπου μακριά --την Αγγλία, την Αίγυπτο, τη Σοβιετική Ενωση-- εκπομπές που σε άλλες συνθήκες θα χάνονταν στα παράσιτα.

Νοσταλγία, αγαπητοί μου, νοσταλγία.

Αλλά και πίκρα.



Την εποχή εκείνη, των μόντεμ με τα 1200bps, του gopher και του lynx, φανταζόμασταν ένα τεχνολογικό μέλλον αδελφωμένο και φωτεινό, όπου η πληροφορία θα ήταν ελεύθερη και η γνώση προσιτή.

Τα μόνα εμπόδια ήταν, στα μάτια μας, τα τεχνολογικά. Μόνο αυτά μπορούσαμε να δούμε. Μόνο αυτά υπήρχαν. Όμως θυμόμασταν το νόμο του Moore και ξέραμε, ήμασταν σίγουροι, πως επρόκειτο για εμπόδια πρόσκαιρα.

Τελείως πρόσκαιρα: Το 1993, που βγήκε το Mosaic σε beta, είχα ήδη αποκτήσει μόντεμ που έτρεχε στα 14400bps. Λίγα μόλις χρόνια πιο μετά και το ερώτημα της ταχύτητας σύνδεσης δεν απασχολούσε πλέον κανέναν -- και πάντως όχι ως εμπόδιο.

Σήμερα η πληροφορία είναι πλέον ελεύθερη (ή σχεδόν ελεύθερη), και η γνώση προσιτή.

Το μέλλον είναι ήδη εδώ, όπως το φανταζόμασταν. Τεχνολογικό είναι, εντάξει. Αδελφωμένο είναι; Φωτεινό;

Φωτεινό; Ναι καλά.

Κοιτάχτε, μπορεί να σας γράφω με τη γραφομηχανή, αλλά δεν έχω ξεκόψει και τελείως από το ίντερνετ. Δανείζομαι τον φορητό της εξαδέλφης. Μπαίνω στο Facebook, κοιτάω τις συζητήσεις και τα σχόλια, ρίχνω ματιές στα blog.

Διαβάζω και αναστατώνομαι. Τέτοιο μίσος, τέτοια χολή, και τέτοια βλακεία δεν τα χωράει ο νους μου. Με ξεπερνάνε.

Ναι, η πληροφορία είναι πλέον ελεύθερη.

Αλλά τι να την κάνεις την ελεύθερη πρόσβαση σε όλες τις βιβλιοθήκες του κόσμου, κακόμοιρε, αν δεν ξέρεις καν να διαβάζεις;

Τι να σου κάνει η γνώση αν δεν έχεις μάθει πρώτα να καταλαβαίνεις;

Τι να το κάνεις το φωτεινό μέλλον, αν είσαι τόσο τυφλωμένος από τη βλακεία που δεν βλέπεις πέρα από τη μύτη σου;

Μελαγχολώ.

Η ελευθερία, αγαπητοί μου, δεν είναι αντικείμενο: Δεν είναι σουπερκινητήλέφωνο (TM) με κάμερα, πυξίδα, παγούρι, και θερμόμετρο, να το παίρνεις μαζί σου στην καφετέρια να κάνεις φιγούρα στα διπλανά τραπέζια. Δεν είναι κονκάρδα να την κοτσάρεις στο σακάκι, κάθε που σου κάνει κέφι.

Η ελευθερία δεν είναι στατική κατάσταση· είναι ορίζοντας, δυνατότητα, διαδικασία --process, για να το πω και στ' αγγλικά.

Η ελευθερία δεν είναι καν δικαίωμα. Είναι καθήκον. Και ως τέτοιο θέλει «αρετήν και τόλμην».

Τότε μόνο γίνεσαι ελεύθερος, όταν είσαι έτοιμος να το παλέψεις και να κουραστείς. Αλλιώς η «ελευθερία» σου δεν είναι παρά ένα ακόμα βαρίδι που σε κρατάει βυθισμένο στην αμάθεια, δέσμιο της βλακείας σου, υποδουλωμένο.

Σας έγραφα την άλλη φορά, αγαπητοί μου, για το ξυράφι του Όκκαμ. Ελπίζω πως το ψάξατε.

Ελπίζω να έχετε πλέον καταλάβει πως πρόκειται για ξυράφι, και όχι για μαγικό ραβδί. Και ότι πρέπει να το χρησιμοποιήσετε. Πώς;

Δανείζομαι την προτροπή του Καντ:

Sapere aude!

(Γουγκλ ιτ.)

Σας ασπάζομαι,
Θείος Ακάκιος


Τα επεισόδια του deltaCast, μοιάζουν με ταινίες.
Στη διάρκεια -- όχι στην ποιότητα.
deltaCast s02e01
OpenVAS vScanner
http://deltahacker.gr/category/deltacast

Skill: Beginner
Tags: Raspberry Pi, FM, e-radio, sox, stream

by multiPetros



Δώστε διαδικτυακή φωνή στο παλιό τρανζιστοράκι!

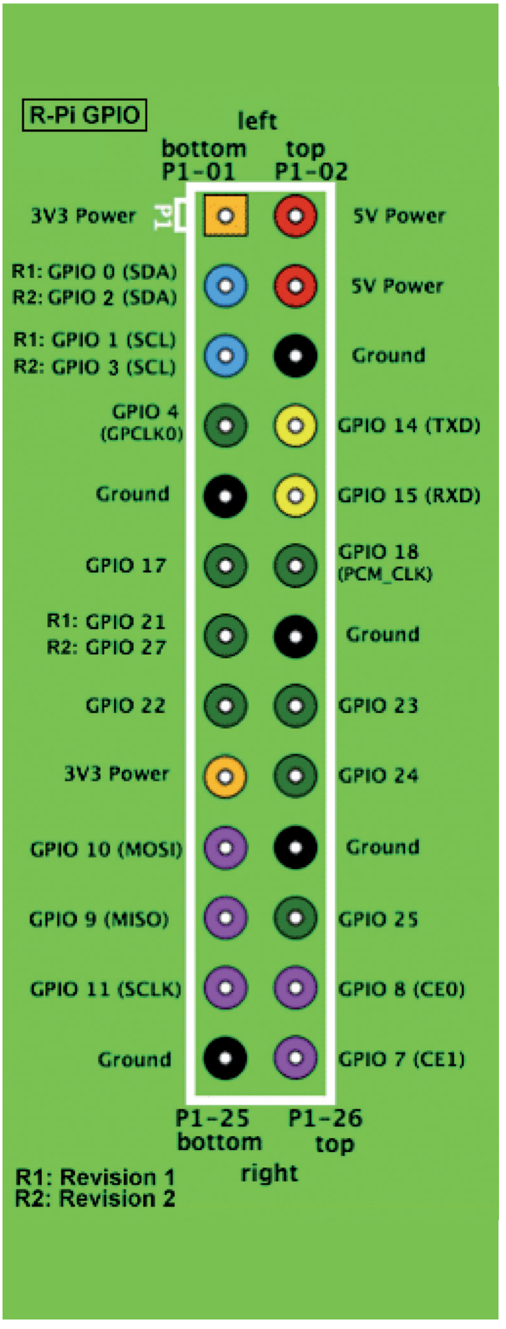
Μέρα με τη μέρα, το Raspberry Pi κερδίζει την καρδιά μας όλο και περισσότερο. Ο μικρός υπολογιστής των 30 δολαρίων μπορεί να δώσει περισσότερες λύσεις απ' όσες φανταζόμαστε. Πρόσφατα, σε κάποιο εργαστήριο της Parabing Creations, κατάφερε να ζωντανέψει ένα από 'κείνα τα παλιά κι αγαπημένα ραδιόφωνα, δίνοντάς του φωνή από το Internet. Σύμμαχός του ήταν μόνο ένα μικρό κομμάτι χαλκού και μερικές εκατοντάδες γραμμές κώδικα...

Στη δυτική όχθη του Ατλαντικού βρίσκονται κολοσσοί της Πληροφορικής κι από 'κεί έρχονται και τα περισσότερα τεχνολογικά θαύματα. Ωστόσο, η Αγγλία έχει αποδείξει πολλές φορές ότι η δική μας όχθη δεν υστερεί σε τίποτα. Κατά περιόδους έχουν υπάρξει πολλές αγγλικές εταιρείες με προϊόντα-διαμάντια, που έχουν αφήσει ανεξίτηλο στίγμα στο χώρο της Πληροφορικής. Ποιος δεν γνωρίζει ή δεν έχει ακούσει για τις Acorn, Amstrad και Sinclair;[1] Θα τολμούσαμε να πούμε ότι η παράδοση αυτή συνεχίζεται μέχρι σήμερα, με το Raspberry Pi Foundation και το πνευματικό του παιδί, το πολυτάλαντο Raspberry Pi. Οι χρήσεις που μπορεί να βρει αυτός ο λιλιπούτειος κι αθόρυβος υπολογιστής είναι ατελείωτες. Εμείς πάντως, τον αξιοποιήσαμε πρόσφατα για να δώσουμε νέα ζωή σε ένα παμπάλαιο ραδιοφωνάκι. Με τη βοήθειά του, το ραδιοφωνάκι μας παίζει οποιονδήποτε δικτυακό ραδιοφωνικό σταθμό θέλουμε και όπως καταλαβαίνετε, έχει τέλεια λήψη παντού[2] ;)

Τοπικός ραδιοφωνικός σταθμός

Αναρωτιέστε πώς τα καταφέραμε; Ελπίζουμε να μην τρέχει ο νους σας σε κάποια βάρβαρη επέμβαση στα ηλεκτρονικά του Raspberry Pi ή του ραδιοφώνου. Δεν τροποποιήσαμε κανένα κύκλωμα κι ούτε του φυτέψαμε στο ραδιόφωνο κάποια είσοδο line-in. Η λύση που αξιοποιήσαμε δεν ήταν δικής μας έμπνευσης, αλλά είναι σίγουρα πανέξυπνη.

Η πλακέτα του RasPi διαθέτει μια σειρά 26 ακροδεκτών, που είναι γνωστοί σαν GPIO[3]. Οι εν λόγω ακροδέκτες επιτρέπουν την επικοινωνία, σε χαμηλό επίπεδο, μεταξύ του Raspberry και διαφόρων εξωτερικών συσκευών ή ακόμα και των δικών μας κυκλωμάτων. Ουσιαστικά,



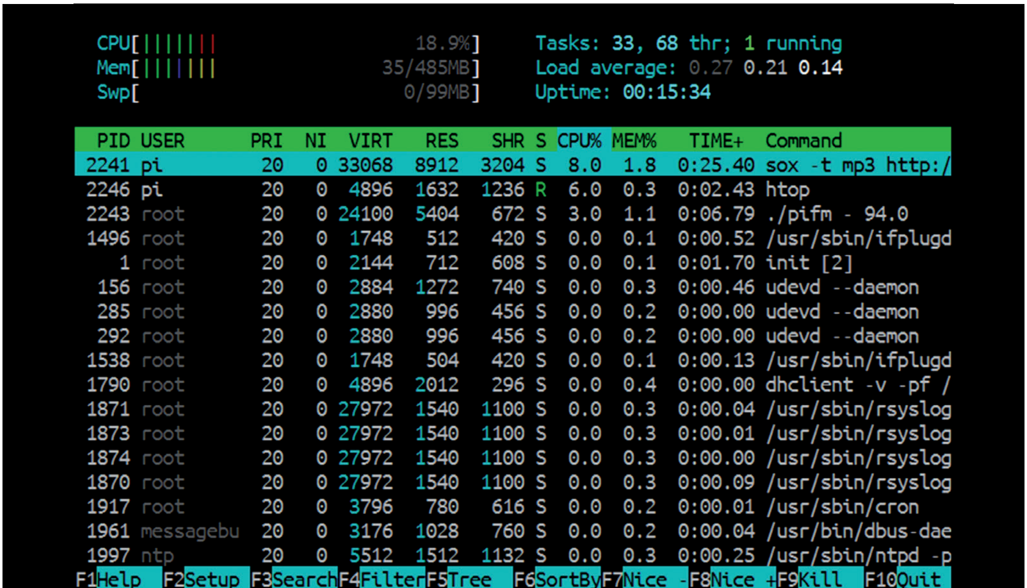
Σχεδιάγραμμα απαρίθμησης των ακροδεκτών GPIO του Raspberry Pi.

1. Αν για κάποιο λόγο ανήκετε στη μειονότητα εκείνων που δεν γνωρίζουν αυτές τις εταιρείες, αξίζει να κάνετε μια βόλτα από το κανάλι του φίλου και συνεργάτη Sonic, στο YouTube: http://youtube.com/user/sonic2000gr/videos.
2. Για να 'μαστε ειλικρινείς, το ραδιοφωνάκι μας έχει τέλεια λήψη οπουδήποτε έχουμε και τέλειο σήμα Internet.
3. General Purpose Input/Output

μπορούν να στείλουν οποιαδήποτε αλληλουχία ψηφιακών παλμών και σχεδόν με οποιαδήποτε συχνότητα[4]. Αυτή τη φαινομενικά άσχετη με το θέμα μας δυνατότητα αξιοποίησαν οι Oliver Mattos και Oskar Weigl. Οι δύο ευφυέστατοι προγραμματιστές χρησιμοποίησαν τη γλώσσα C και τις δυνατότητες που προσφέρει η πλατφόρμα του Raspberry Pi, ώστε να δημιουργήσουν πάνω σε κάποιον ακροδέκτη GPIO ένα σήμα "παρόμοιο" με τα ραδιοφωνικά σήματα της μπάντας των FM! Επιπρόσθετα, προς μεγάλη χαρά όλων, ανέπτυξαν περαιτέρω το λογισμικό τους, το έκαναν ιδιαίτερα φιλικό για τον απλό χρήστη κι έδωσαν τη δυνατότητα εκπομπής ενός αρχείου ήχου, σε οποιαδήποτε ραδιοφωνική συχνότητα! Με λίγα λόγια, ανέπτυξαν μια εφαρμογή η οποία μετατρέπει το Raspberry Pi σε έναν (μικρής ισχύος) ραδιοφωνικό σταθμό. Το σχετικό project ονομάζεται PiFM και είναι διαθέσιμο υπό τους όρους χρήσης της άδειας GNU GPL. Καλό αυτό.

Η δική μας εφαρμογή

Αποφασίσαμε να χρησιμοποιήσουμε το project PiFM για να μετατρέψουμε το Raspberry Pi σ' έναν πομπό κυμάτων FM (FM transmitter) οικιακού βεληνεκούς. Για αυτό το σκοπό επιστρατεύσαμε το πρόγραμμα SoX, που το δίχως άλλο αποτελεί τον ελβετικό σουγιά για το transcoding του ήχου. Με τη βοήθειά του SoX τροφοδοτήσαμε το PiFM με το audio stream του αγαπημένου μας δικτυακού σταθμού — και μάλιστα σε πραγματικό χρόνο. Έτσι, καταφέραμε να ακούμε τους σταθμούς από σχεδόν οποιοδήποτε δωμάτιο του σπιτιού και μέσα από τα παλιά, αναλογικά ραδιοφωνάκια! Και γιατί να το κρύψουμε, άλλωστε; Με αυτόν τον τρόπο μπορέσαμε ν' ακούμε την αγαπημένη μας μουσική ακόμα και μέσα στο μπάνιο, με το παλιομοδίτικο (αλλά σούπερ ανθεκτικό στους υδρατμούς) τρανζιστοράκι!



Τόσο το PiFM όσο και το SoX, επιβαρύνουν ελάχιστα τον επεξεργαστή του Raspberry Pi. Έτσι, το σύστημά μας θα μπορεί να χρησιμοποιείται ταυτόχρονα και για άλλες δουλειές.

4. Προφανώς, η ταχύτητα του Raspberry Pi και του κυκλώματος ελέγχου των ακροδεκτών θέτει ένα άνω όριο στην ταχύτητα "εκπομπής" των παλμών.

Ώρα για δράση

Ας αφήσουμε όμως τη θεωρία κι ας σηκώσουμε τα μανίκια, για να ξεκινήσουμε τη δημιουργία του οικιακού μας ραδιοφωνικού σταθμού. Κατ' αρχάς χρειαζόμαστε το πρόγραμμα PiFM, το οποίο μπορούμε να κατεβάσουμε από τη διεύθυνση http://omattos.com/pifm.tar.gz. Αυτό γίνεται απευθείας από το Raspberry Pi, χρησιμοποιώντας το δημοφιλές εργαλείο wget:

```
wget http://omattos.com/pifm.tar.gz
```

Μόλις ολοκληρωθεί η λήψη του αρχείου (έχει μέγεθος περί τα 5,5MB), πρέπει να αποσυμπιέσουμε το περιεχόμενό του με το εργαλείο tar:

```
tar -xvf pifm.tar.gz
```

Το συμπιεσμένο πακέτο περιλαμβάνει τόσο τον πηγαίο κώδικα, όσο και το προϊόν της μεταγλώττισής του (δηλαδή το πρόγραμμα PiFM). Το μόνο που έχουμε να κάνουμε εμείς είναι να δώσουμε την ιδιότητα του εκτελέσιμου στο εν λόγω αρχείο. Για το σκοπό αυτό, χρησιμοποιούμε το cmod κάπως έτσι:

```
cmod +x pifm
```

Παρεμπιπτόντως, αν σας προβληματίζει η πιθανή επιβάρυνση που ενδέχεται να επιφέρει στο σύστημα η λειτουργία του PiFM, πρέπει να πάψετε να ανησυχείτε. Ο κώδικάς του λειτουργεί αψεγάδιαστα και χρησιμοποιεί κλήσεις DMA, ώστε να απασχολεί τον επεξεργαστή ελάχιστα. Στη χειρότερη περίπτωση, θα διαπιστώσετε ότι καταλαμβάνει μόλις το 3% του επεξεργαστή.

Χρήση του PiFM

Ο τρόπος χρήσης της νέας μας εφαρμογής χαρακτηρίζεται από μεγάλη απλότητα. Αρκεί να το εκτελέσουμε με δικαιώματα υπερχρήστη[5] (με τη βοήθεια του sudo) και να του δώσουμε δύο μόλις παραμέτρους: Η πρώτη περιγράφει τη διαδρομή προς το αρχείο ήχου που θέλουμε να εκπέμψουμε και η δεύτερη προσδιορίζει την επιθυμητή συχνότητα εκπομπής. Για να πετύχουμε το καλύτερο δυνατό αποτέλεσμα, πρέπει να εξασφαλίσουμε ότι η συχνότητα εκπομπής που θα επιλέξουμε είναι ελεύθερη ή, τέλος πάντων, περιέχει μόνο ασθενή σήματα που θα υπερκαλύψουμε άνετα με τον δικό μας πομπό. Προκειμένου να εντοπίσουμε μία "άδεια" συχνότητα, αρκεί να σαρώσουμε την μπάντα των FM με ένα ραδιοφωνάκι και να σταματήσουμε στη συχνότητα που έχει τα περισσότερα παράσιτα και βρίσκεται όσο πιο μακριά γίνεται από σταθμούς με δυνατό σήμα. Επίσης, σε ό,τι αφορά το αρχείο ήχου που θέλουμε να εκπέμψουμε, πρέπει να εξασφαλίσουμε ότι είναι τύπου WAV, με συχνότητα δειγματοληψίας στα 22.5kHz και "βάθος" 16bit.

Για μια πρώτη δοκιμή, μπορούμε να εκτελέσουμε το πρόγραμμα με το αρχείο ήχου που βρήκαμε μέσα στο πακέτο του PiFM και μια τυχαία συχνότητα. Εμείς επιλέξαμε τα 101MHz:

```
sudo ./pifm sound.wav 101.0
```

Σε αυτό το στάδιο, αν δοκιμάσουμε να συντονίσουμε κάποιο ραδιόφωνο στην επιλεγμένη συχνότητα, είναι 97.6361% σίγουρο ότι δεν θα ακούσουμε τίποτα. Αυτό οφείλεται στο μικροσκοπικό μέγεθος της κεραίας του πομπού μας. Βλέπετε, στο ρόλο της κεραίας βρίσκεται απλά ο ακροδέκτης GPIO 4. Αυτό που μένει ώστε ο

5. Αυτό είναι υποχρεωτικό, ώστε το πρόγραμμα να έχει την απαιτούμενη πρόσβαση στο υλικό του Raspberry Pi.

πομπός μας να είναι πλήρως λειτουργικός, είναι να συνδέσουμε ένα μικρό χάλκινο καλώδιο (5cm είναι αρκετά) στον συγκεκριμένο ακροδέκτη. Ο GPIO 4, γνωστός και ως GPCLK0, είναι ο ακροδέκτης υπ' αριθμόν 7. Για να τον εντοπίσουμε, πρέπει να ξεκινήσουμε την απαρίθμηση των ακροδεκτών από τα αριστερά. Οι ακροδέκτες είναι διατεταγμένοι σε δύο σειρές. Όσοι βρίσκονται στην πάνω σειρά έχουν ζυγό αριθμό, ενώ αυτοί που βρίσκονται στην κάτω έχουν περιττό. Δείτε την σχετική εικόνα του άρθρου και θα καταλάβετε αμέσως. Για τη σύνδεση του ακροδέκτη και του μικρού καλωδίου, μπορούμε να χρησιμοποιήσουμε ένα jumper, σαν αυτά που συναντάμε στις μητρικές. Μετά την προσθήκη της κεραίας, το σήμα του πομπού μας θα αρχίσει ν' ακούγεται πεντακάθαρα.

Προκειμένου να σταματήσουμε την αναπαραγωγή, σταματάμε την εκτέλεση του προγράμματος πατώντας το συνδυασμό πλήκτρων [Ctrl+C]. Όπως καταλαβαίνετε, ο καλύτερος τρόπος για να χρησιμοποιούμε το PiFM χωρίς να δεσμεύουμε μία ολόκληρη κονσόλα, είναι με τη βοήθεια του εργαλείου screen.

Ραδιοφωνικό σήμα

Αφού υλοποιήσουμε το πρώτο και βασικό τμήμα της κατασκευής μας —τον πομπό—, μπορούμε να προχωρήσουμε στη χρήση του SoX. Όπως είπαμε και παραπάνω, με τη βοήθειά του θα τροφοδοτήσουμε τον πομπό μας με το σήμα κάποιου δικτυακού ραδιοφωνικού σταθμού. Ξεκινάμε με την εγκατάσταση του SoX η οποία δεν θα μας δυσκολέψει καθόλου, αφού το πρόγραμμα διατίθεται μέσα από το repository του Raspbian:

```
sudo apt-get install sox libsox-fmt-all
```

Το πρόσθετο πακέτο που εγκαθιστούμε (libsox-fmt-all) περιέχει κάποιες απαραίτητες για την εργασία μας βιβλιοθήκες. Στη συνέχεια, αρκεί να εντοπίσουμε το URL του stream του αγαπημένου μας σταθμού. Το πώς θα το βρούμε αυτό, διαφέρει κατά περίπτωση. Σε γενικές γραμμές, μπορούμε να ακολουθήσουμε την εξής μεθοδολογία: Μεταβαίνουμε στο site του σταθμού, ανοίγουμε την ειδική σελίδα με τον web player κι αρχίζουμε να μελετάμε το πηγαίο της κώδικα. Το ζητούμενο URL θα βρίσκεται κάπου εκεί μέσα. Στην περίπτωσή μας, από το site του RSO βρήκαμε ότι το σχετικό URL είναι το `http://rso.isolservers.com:8200`. Βέβαια, οι δικτυακοί ραδιοφωνικοί σταθμοί δεν εκπέμπουν τον ήχο σε WAV, αλλά σε MP3. Εδώ μπαίνει στο παιχνίδι το SoX, που θα κάνει την κατάλληλη μετατροπή.

Όπως αντιλαμβάνεστε, ούτε χρειαζόμαστε ούτε θέλουμε να δημιουργήσουμε ένα τεράστιο αρχείο που θα περιλαμβάνει τη ραδιοφωνική εκπομπή. Γι' αυτό το λόγο θα διασωληνώσουμε την έξοδο του SoX με την είσοδο του PiFM. Δείτε τι κάναμε εμείς, ώστε να εκπέμψουμε το σήμα του RSO με τον πομπό μας:

```
sox -t mp3 http://rso.isolservers.com:8200/ -t wav --input-buffer 80000 -b
16 -r 22050 -c 1 /dev/stdout | sudo ./pifm /dev/stdin 94.0
```

Αρχικά ορίζουμε την κωδικοποίηση του σήματος εισόδου (mp3) και φυσικά την πηγή του σήματος (το URL με το σήμα του σταθμού). Ακολούθως περιγράφουμε τη μορφή που θέλουμε να αποκτήσει το σήμα μας. Σαν έξοδο για το SoX επιλέγουμε την τυπική έξοδο (/dev/stdout). Στη συνέχεια χρησιμοποιούμε τη διασωλήνωση, ώστε η έξοδος του SoX να διοχετευτεί στο PiFM. Όπως βλέπετε, κατά την εκτέλεση του προγράμματος του πομπού, δεν επιλέγουμε κάποιο απλό αρχείο σαν είσοδο αλλά την τυπική είσοδο (stdin).

Με την ευκαιρία, αξίζει να αναφέρουμε και μια εναλλακτική σύνταξη που υποστηρίζουν ορισμένα προγράμματα του Linux, ώστε να πληκτρολογούμε λιγότερο. Αντί να χρησιμοποιήσουμε τα /dev/stdout και /dev/stdin, μπορούμε στη θέση τους να βάλουμε από μια απλή παύλα! Η γραμμή που είδαμε παραπάνω θα μπορούσε να γραφτεί κι ως εξής:

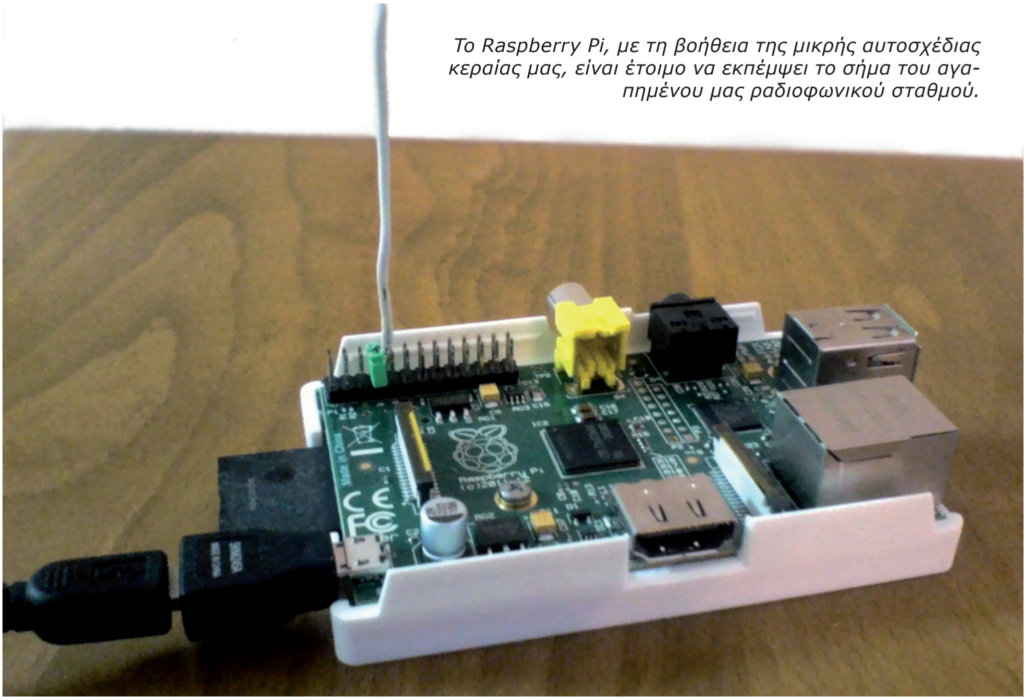
```
sox -t mp3 http://rso.isolservers.com:8200/ -t wav --input-buffer 80000 -b
16 -r 22050 -c 1 - | sudo ./pifm - 94.0
```

Σημειώστε ότι αυτή η σύμβαση δεν αναγνωρίζεται από το κέλυφος του συστήματος, αλλά από συγκεκριμένα μόνο προγράμματα. Επομένως, θα ήταν φρόνιμο να μην την υιοθετήσετε συνολικά και να μην την χρησιμοποιείτε οπουδήποτε.

Επίλογος

Δεν ξέρουμε αν το καταλάβατε, αλλά μόλις τελειώσαμε. Ο στόχος μας επιτεύχθηκε! Αν ανοίξουμε το ραδιόφωνο και το συντονίσουμε στη συχνότητα που έχουμε επιλέξει, θ' ακούσουμε το πρόγραμμα του αγαπημένου μας ιντερνετικού σταθμού :D

Πέρα από την τεχνολογική διάσταση του εγχειρήματος, υπάρχει και η νομική. Οι συχνότητες των FM δεν διατίθενται προς χρήση ελεύθερα και για την εκπομπή σήματος απαιτείται άδεια από την αρμόδια Αρχή. Φυσικά, όταν εκπέμπουμε σε πολύ χαμηλή ισχύ και το σήμα μας περιορίζεται πρακτικά εντός του σπιτιού, δεν πρόκειται να μας ασκήσει κανείς δίωξη. Σε κάθε περίπτωση, καλό θα ήταν να έχουμε υπόψη μας το ζήτημα και να προσέχουμε πώς χρησιμοποιούμε την κατασκευή. Αν μπείτε στον πειρασμό να συνδέσετε την έξοδο του Raspberry Pi με κάποιον ενισχυτή σημάτων FM κι ενδεχομένως με καλύτερη κεραία, ενδέχεται να γίνετε σύγχρονοι ραδιοφωνικοί πειρατές ;)



Το Raspberry Pi, με τη βοήθεια της μικρής αυτοσχέδιας κεραίας μας, είναι έτοιμο να εκπέμψει το σήμα του αγαπημένου μας ραδιοφωνικού σταθμού.

Skill: Beginner
Tags: database, SQL, query, website engine, sanitization, escaping

by Sp1r@lEv0lut10n



Ύπουλη εισβολή, κανονικά και με το νόμο!

Στήνεις το δικό σου site, το δημοσιεύεις σε δικό σου server για να έχεις τον έλεγχο, φροντίζεις να ενημερώνεις το λειτουργικό σύστημα του server τακτικά, εγκαθιστάς κάθε patch που κυκλοφορεί για τον Apache και νομίζεις πως είσαι ασφαλής. Μπορεί με όλα τα παραπάνω να κουράζεσαι, ωστόσο σκέφτεσαι ότι με τόσο κόπο και προσπάθεια είναι αδύνατο να κινδυνεύεις. Στην πραγματικότητα, όμως, γελιέσαι. Αν πρόκειται για ένα σύγχρονο site, είναι βέβαιο ότι η λειτουργία του θα στηρίζεται σε κάποια βάση δεδομένων. Άραγε πόσο ασφαλής είναι η επικοινωνία μεταξύ της μηχανής ενός site και της εκάστοτε βάσης δεδομένων; Μήπως υπάρχει μπροστά στα μάτια μας μία ακόμα ορθάνοιχτη πόρτα για τους επίδοξους εισβολείς;
Μα, τι σημαίνουν όλα αυτά τέλος πάντων;

Φανταστείτε ένα μηχανισμό που επιτρέπει τη δημιουργία και τη διαχείριση πινάκων με οποιουδήποτε είδους στοιχεία (αριθμούς, σύμβολα, γράμματα κ.ο.κ.). Ένας τέτοιος μηχανισμός ονομάζεται βάση δεδομένων (database). Οι βάσεις δεδομένων μπορούν να συγκροτούν αυτόματα νέους πίνακες, συνδυάζοντας ολόκληρες στήλες από ξεχωριστούς πίνακες. Μπορούν επίσης να προσθέτουν ή να αφαιρούν γραμμές από πίνακες και, το σημαντικότερο, μπορούν να πραγματοποιούν οσοδήποτε σύνθετες αναζητήσεις θέλουμε. Γι' αυτή την τελευταία εργασία οι βάσεις δεδομένων ενσωματώνουν μία εξειδικευμένη γλώσσα, με τη βοήθεια της οποίας ο χρήστης μπορεί να καθορίζει τα κριτήρια αναζήτησης. Από την αποστολή της γλώσσας προκύπτει και το όνομά της, που είναι το Structured Query Language (Δομημένη Γλώσσα Ερωτημάτων) ή πιο απλά SQL (προφέρεται "σίκουελ"). Βέβαια, η εν λόγω γλώσσα δεν επιτρέπει μόνο το σχηματισμό ερωτημάτων. Επιτρέπει κι όλες τις παρεμφερείς εργασίες που περιγράψαμε παραπάνω (συνένωση πινάκων, τροποποίηση δεδομένων, διαγραφή γραμμών ή κελιών κ.ο.κ). Το συντακτικό της SQL είναι πολύ απλό και θυμίζει εκείνο των αγγλικών, ενώ το λεξιλόγιό της είναι περιορισμένο και περιλαμβάνει απλές λέξεις. Διαβάζοντας μία πρόταση σε SQL είναι βέβαιο ότι θα καταλάβουμε αμέσως το νόημά της, ακόμα κι αν βλέπουμε τη γλώσσα για πρώτη φορά. Μερικές δημοφιλείς βάσεις δεδομένων είναι οι Oracle, MySQL[1], PostgreSQL, Microsoft Access κ.ά. Εδώ πρέπει να υπογραμμίσουμε ότι κάθε υλοποίηση υποστηρίζει τη δική της εκδοχή για τη γλώσσα SQL, αν και σε όλες τις παραλλαγές υπάρχει ένα κοινό σύνολο βασικών/θεμελιωδών λέξεων. Ας δούμε δύο απλά παραδείγματα χρήσης της γλώσσας SQL:

username	password	id	settings	msg_id
Panos	12e45	73	yes, no, no, no	26, 31, 41, 15, 56, 5, 44, 17, 62
Tasos	t6712sa	84	no, no, no, no	15, 4, 62, 114, 82, 52, 12
dimitris	d1m1990	100	no, yes, yes, no	9, 20, 38, 55
giorgos	G1@nn00g	82	yes, no, yes, no	20, 31, 16, 22, 2, 74, 51, 3, 6, 7, 91
Admin	2u5%1s3	1	yes, yes, yes, yes	0

```
SELECT settings FROM users WHERE username = 'panos' AND password = '12e45';
SELECT * FROM users WHERE user_id = '10' OR user_id = '100';
```

Παραπάνω βλέπουμε έναν πίνακα με το όνομα users. Οι δύο προτάσεις που ακολουθούν είναι γραμμένες στη γλώσσα SQL και, για την ακρίβεια, με το συντακτικό που υποστηρίζει η βάση δεδομένων MySQL. Οι προτάσεις αυτού του είδους ονομάζονται και ερωτήματα (queries). Το πρώτο ερώτημα επιστρέφει τις ρυθμίσεις (settings) για το χρήστη που έχει όνομα (username) το panos και συνθηματικό (password) το 12345. Το δεύτερο ερώτημα επιστρέφει ολόκληρες τις γραμμές που αφορούν στους χρήστες με αναγνωριστικό κωδικό (id) το 10 ή το 100. Απλό δεν είναι;

1. Αποτελεί ιδιοκτησία της Oracle!

Βασικά, οι βάσεις υπάρχουν παντού

Πίσω από κάθε σύγχρονο site βρίσκεται μια βάση δεδομένων. Εκεί διατηρείται όλο το περιεχόμενο του εκάστοτε δικτυακού τόπου (άρθρα, κατηγορίες άρθρων, σχόλια, πληροφορίες για τις εικόνες κ.λπ.). Πλέον, στα περισσότερα sites συναντάμε και την έννοια του λογαριασμού χρήστη. Έτσι, υπάρχει η ανάγκη για έναν μηχανισμό που θα διατηρεί τα στοιχεία των χρηστών (username και password), καθώς και τις όποιες επιλογές/ρυθμίσεις έχει πραγματοποιήσει καθένας. Όπως υποψιάζεστε, για αυτή την εργασία χρησιμοποιείται η βάση δεδομένων, στην οποία διατηρείται και ολόκληρο το περιεχόμενο του site! Με λίγα λόγια, όλα τα δεδομένα που διατηρούνται σε έναν σύγχρονο δικτυακό τόπο, βρίσκονται αποθηκευμένα στους πίνακες μιας βάσης δεδομένων. Μήπως έχετε αρχίσει να σκέπτεστε πονηρά;

Αδύναμος κρίκος

Οι δικτυακές τοποθεσίες που χρησιμοποιούν κάποια βάση δεδομένων απαρτίζονται από σελίδες γραμμένες σε HTML. Σε αυτές τις σελίδες όμως δεν περιλαμβάνεται το περιεχόμενο, παρά μόνο η δομή με την οποία πρόκειται να προβληθεί. Το περιεχόμενο "φυτεύεται" μέσα στη δομή που περιγράφει ο κώδικας HTML, με τη βοήθεια προγραμμάτων (scripts) που γράφονται σε διάφορες γλώσσες προγραμματισμού (PHP, Ruby, Python κ.ά.). Αυτά τα scripts αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων και μεταφέρουν από και προς αυτήν τα απαιτούμενα δεδομένα, χρησιμοποιώντας τα κατάλληλα ερωτήματα (queries). Τα εν λόγω προγράμματα αποτελούν αυτό που ονομάζουμε "μηχανή" (website engine) και είναι ο αδύναμος κρίκος στην ασφάλεια του site! Ένα script αυτού του είδους που γράφτηκε απρόσεκτα, είναι πιθανό να επιτρέπει το λεγόμενο SQL injection. Αυτή η ευπάθεια (vulnerability) αναφέρεται στη δυνατότητα του επιτιθέμενου να αλλοιώσει τα queries που αποστέλλονται στη βάση δεδομένων. Κάτι τέτοιο προϋποθέτει την παράνομη εισαγωγή κώδικα (εδώ κολλάει το injection), γραμμένου σε SQL. Μήπως όλα αυτά μοιάζουν ακατάληπτα; Ας δούμε ένα παράδειγμα...

Η βασική ιδέα

Φανταστείτε ένα site που προσφέρει μια υπηρεσία τύπου webmail. Στην αρχική σελίδα, πέρα από οτιδήποτε άλλο, εμφανίζεται ένα πεδίο για την εισαγωγή του ονόματος χρήστη (username) κι ένα ακόμα για την εισαγωγή του αντίστοιχου συνθηματικού (password). Όταν κάποιος επισκέπτης συμπληρώνει τα πεδία και κάνει κλικ στο κουμπάκι login, sign-in ή όπως αλλιώς το έχουν ονομάσει, η μηχανή του site ελέγχει αν το ζεύγος username-password υπάρχει στη βάση δεδομένων. Αν το αποτέλεσμα του ελέγχου είναι θετικό —αν δηλαδή υπάρχει στη βάση το username κι αν του έχει αντιστοιχηθεί το δοσμένο password— η μηχανή του site θεωρεί ότι η απόπειρα σύνδεσης ήταν επιτυχημένη. Στη συνέχεια, φορτώνει τα μηνύματα του σχετικού λογαριασμού, τα παρουσιάζει και προχωρά σε όποιες άλλες ενέργειες σχετίζονται με τη σύνδεση του χρήστη. Στην αντίθετη περίπτωση —αν το username δεν υπάρχει στη βάση ή αν ο αντίστοιχος κωδικός είναι διαφορετικός από αυτόν που έδωσε ο χρήστης—, εμφανίζεται ένα μήνυμα αποτυχίας και η μηχανή του site ζητά από τον επισκέπτη να προσπαθήσει εκ νέου. Ε, λοιπόν, ο μηχανισμός που περιγράψαμε θα μπορούσε να πραγματοποιηθεί με τη βοήθεια ενός απλού ερωτήματος προς τη βάση:

```
SELECT * FROM users WHERE username = 'pvar' AND password = '123456';
```

Το παραπάνω ερώτημα ελέγχει αν υπάρχει ο χρήστης pvar καθώς κι αν του αντι-

στοιχεί ο κωδικός 123456. Εφόσον επαληθεύονται τα κριτήρια αναζήτησης, η βάση θα επιστρέψει όλα τα δεδομένα που αντιστοιχούν στο χρήστη pvar. Αν η βάση δεν επιστρέψει τίποτα, η μηχανή του site συμπεραίνει ότι δεν υπάρχει τέτοιος λογαριασμός χρήστη ή ότι ένα από τα δύο δοσμένα στοιχεία (το username ή το password) είναι λανθασμένο. Νομίζουμε ότι η λογική με την οποία λειτουργεί το πρόγραμμα πίσω από το site είναι ξεκάθαρη. Ωστόσο, εδώ υπάρχει μια σημαντική λεπτομέρεια: Το ερώτημα που θα ασταλεί στη βάση πρέπει να σχηματίζεται από τη μηχανή του site και φυσικά, πρέπει να περιέχει τα στοιχεία που έδωσε ο επισκέπτης. Αν αυτό το τμήμα του κώδικα είναι γραμμένο με προχειρότητα ή με αφέλεια, το site θα είναι ευπαθές σε επιθέσεις SQL injection. Και για να γίνουμε ακόμα πιο συγκεκριμένοι, ας δούμε πώς θα μπορούσαν να σχηματίζονται τα ερωτήματα προς τη βάση. Η γραμμή κώδικα που ακολουθεί είναι γραμμένη σε PHP και νομίζουμε ότι είναι κατανοητή:

```
$dbquery = "SELECT * FROM users WHERE username='" . $form_usr . "' AND
password= '" . $form_pwd . "';"
```

Η παραπάνω γραμμή σχηματίζει το ερώτημα με τη μορφή ενός string και το αποθηκεύει στη μεταβλητή &dbquery. Το string με το ερώτημα σχηματίζεται από τρία σταθερά τμήματα, στα οποία προστίθενται οι τιμές των μεταβλητών $form_usr και $form_pwd. Οι τελευταίες δύο μεταβλητές θεωρούμε ότι περιέχουν το όνομα χρήστη και το συνθηματικό που έδωσε ο επισκέπτης στα σχετικά πεδία. Μέχρι στιγμής είναι όλα απλά και λογικά. Πράγματι, μόνο που αν οι μεταβλητές $form_usr και $form_pwd δεν δέχονται το κατάλληλο φιλτράρισμα, το site θα είναι ξέφραγο αμπέλι! Βλέπετε, χωρίς τον κατάλληλο έλεγχο για το περιεχόμενο όσων εισάγει ο επισκέπτης, είναι πιθανό να σχηματιστούν ερωτήματα προς τη βάση με πολύ διαφορετικό νόημα από αυτό που θα ήθελε ο δημιουργός του. Για να φανεί αυτός ο κίνδυνος, θα εξετάσουμε μερικά σενάρια. Σε καθένα από αυτά θεωρούμε ότι ο επισκέπτης του site εισάγει διαφορετικά δεδομένα και οι μεταβλητές με τις αντίστοιχες τιμές δεν ελέγχονται καθόλου.

1. Ο επισκέπτης μας είναι εντελώς αθώος ή, τέλος πάντων, δεν έχει καμία κακή πρόθεση. Έτσι, δίνει το username και το password προσπαθώντας να συνδεθεί στο λογαριασμό του:

   ```
   $form_usr = panos
   $form_pwd = 77634
   ```

   Μ' αυτά τα δεδομένα, το ερώτημα προς τη βάση θα αποκτήσει την ακόλουθη μορφή:

   ```
   SELECT * FROM users WHERE username = 'panos' AND password = '77634';
   ```

   Αν υπάρχει ο χρήστης panos και το συνθηματικό του είναι όντως το 77634, η βάση θα επιστρέψει όλα τα σχετικά δεδομένα από τον πίνακα users. Η μηχανή του site θα πάρει τα δεδομένα, θα τα ενσωματώσει σε κάποια σελίδα HTML και θα τα παρουσιάσει στον επισκέπτη. Φυσικά, αν ο χρήστης δεν υπάρχει ή το συνθηματικό είναι λανθασμένο, το ερώτημα προς τη βάση δεν θα επιστρέψει τίποτα. Έτσι, η μηχανή του site θα θεωρήσει ότι τα δεδομένα που εισήγαγε ο επισκέπτης ήταν λανθασμένα και θα του ζητήσει να προσπαθήσει ξανά.

2. Ο επισκέπτης είναι υποψιασμένος και έχει αρχίσει να παίζει επικίνδυνα. Αντί να συμπληρώσει ένα σκέτο όνομα χρήστη, εισάγει κάτι πιο σύνθετο, που μοιάζει να μην έχει κανένα νόημα. Για συνθηματικό δεν δίνει τίποτα:

```
$form_usr = admin'; --
$form_pwd =
```

Τώρα, το ερώτημα που σχηματίζεται αρχίζει να αποκτά ενδιαφέρον για τον επισκέπτη του site και να γίνεται επικίνδυνο για τον ιδιοκτήτη:

```
SELECT * FROM users WHERE username = 'admin'; -- ' AND password = '';
```

Στη γλώσσα SQL το ελληνικό ερωτηματικό συμβολίζει το τέλος ενός ερωτήματος, ενώ οι δύο διαδοχικές παύλες σηματοδοτούν την αρχή των σχολίων (remarks ή comments). Όπως συμβαίνει σε κάθε γλώσσα προγραμματισμού, έτσι κι εδώ τα σχόλια αγνοούνται παντελώς. Η βάση δεδομένων θα αγνοήσει όλα όσα βρίσκονται μετά τις παύλες (το τμήμα `' AND password = ''`). Τελικά, το ερώτημα που θα φτάσει στη βάση θα ζητά όλα τα δεδομένα του χρήστη admin, χωρίς να πραγματοποιεί κανέναν περαιτέρω έλεγχο. Όπως είναι φανερό, ο επισκέπτης θα μπορούσε να επαναλάβει το ίδιο κολπάκι χρησιμοποιώντας και άλλα πιθανά usernames, όπως τα administrator, root κ.ά. Μ' αυτόν τον τρόπο, αργά η γρήγορα ο επισκέπτης θα μπορέσει να συνδεθεί στο λογαριασμό του διαχειριστή.

3. Ο επισκέπτης αρχίζει να γίνεται πιο πονηρός. Αγνοεί και πάλι το πεδίο για το συνθηματικό, αλλά συμπληρώνει το πεδίο για το username με ακόμα πιο έξυπνο τρόπο:

```
$form_usr = admin' OR 1=1; --
$form_pwd =
```

Οι απανωτές δοκιμές που αναφέραμε προηγουμένως, με διάφορα πιθανά usernames, είναι περιττές: Υπάρχει και πιο σύντομος δρόμος. Με το συγκεκριμένο τρόπο συμπλήρωσης των πεδίων, το ερώτημα προς τη βάση δεδομένων παίρνει την ακόλουθη μορφή:

```
SELECT * FROM users WHERE username = 'admin' OR 1=1; -- "' AND password = '';
```

Η πονηριά του επισκέπτη εντοπίζεται στο τμήμα `OR 1=1`. Ο τελεστής OR καθορίζει ότι αρκεί να ικανοποιείται ένας μόνο από τους δύο ελέγχους (username='admin' και 1=1). Όμως αυτό το 1=1 ικανοποιείται πάντα. Κατά συνέπεια, ο έλεγχος που πραγματοποιεί το παραπάνω ερώτημα είναι πάντοτε αληθής. Έτσι, όταν η βάση δεχτεί το συγκεκριμένο ερώτημα θα δώσει όλα τα δεδομένα που έχουν καταχωρηθεί στον πίνακα users. Βέβαια, ενδέχεται η μηχανή του site να μην είναι σε θέση να διαχειριστεί τα δεδομένα πολλών χρηστών ταυτόχρονα και έτσι να προκύψουν διάφορα σφάλματα κατά τη λειτουργία της. Ωστόσο η βάση δεδομένων θα 'χει μαρτυρήσει τα πάντα και είναι πολύ πιθανό για τον επιτιθέμενο να τα μάθει.

4. Τώρα ο επισκέπτης επιστρατεύει περισσότερες γνώσεις. Αφήνει κενό το πεδίο για το password και στο πεδίο για το username δίνει ένα τυχαίο string, μαζί με μια ολόκληρη φράση σε SQL:

```
$form_usr = xyz' AND 1=(SELECT COUNT(*) FROM test_name ); --
$form_pwd =
```

Αυτή η επίθεση δεν στοχεύει στα αποθηκευμένα δεδομένα αλλά στους πίνακες της βάσης. Το ερώτημα προς τη βάση δεδομένων θα πάρει την ακόλουθη μορφή:

```
SELECT * FROM users WHERE username = 'xyz' AND 1 = (SELECT COUNT(*)
FROM test_name ); -- AND password = '';
```

Η εκτέλεση του `SELECT COUNT(*) FROM test_name` επιστρέφει το πλήθος των γραμμών που περιέχει ο πίνακας test_name. Όμως, είναι σχεδόν απίθανο για έναν πίνακα να περιέχει μία μόνο γραμμή. Άρα, η ισοδυναμία `1 = (SELECT COUNT(*) FROM test_name)` είναι σχεδόν απίθανο να ικανοποιείται. Ας δούμε και τον έλεγχο για το username. Η ισότητα `username = 'xyz'` είναι επίσης απίθανο να ικανοποιείται, εφόσον είναι απίθανο να υπάρχει χρήστης με τέτοιο username. Αναρωτιέστε που αποσκοπεί ο επιτιθέμενος; Αν ο πίνακας test_name υπάρχει, το ερώτημα SQL θα εκτελεστεί κανονικά, οι δύο έλεγχοι θα αποτύχουν και η βάση δεδομένων δεν θα επιστρέψει τίποτα. Έτσι, η μηχανή του site θα απαντήσει με το μήνυμα που εμφανίζεται όταν ο επισκέπτης δίνει λανθασμένα στοιχεία. Αν ο πίνακας test_name *δεν υπάρχει*, η εκτέλεση του ερωτήματος θα αποτύχει και η βάση δεδομένων θα απαντήσει με κάποιο μήνυμα σφάλματος. Σε αυτή την περίπτωση, αν η μηχανή του site είναι όντως κακογραμμένη (κάτι που υποθέσαμε εξαρχής), ο επισκέπτης θα δει ένα μήνυμα σφάλματος που προέρχεται από τη βάση. Έτσι, θα είναι σε θέση να συμπεράνει ότι η βάση *δεν περιλαμβάνει* πίνακα με το όνομα που έδωσε. Επαναλαμβάνοντας αυτή την επίθεση με διαφορετικά ονόματα (π.χ., members, users, user_table κ.ά.) και με τη μέθοδο της εις άτοπον απαγωγής, ο επιτιθέμενος μπορεί τελικά να μάθει ποιους πίνακες περιλαμβάνει η βάση και φυσικά τα αντίστοιχα ονόματα. Στη συνέχεια, αξιοποιώντας αυτή τη γνώση, θα μπορεί να προχωρήσει σε πολύ πιο επικίνδυνες ενέργειες. Αν δεν μας πιστεύετε, δείτε το ακόλουθο σενάριο.

5. Ας υποθέσουμε ότι ο επισκέπτης μαθαίνει το όνομα ενός από τους πίνακες της βάσης, αξιοποιώντας το τέχνασμα που είδαμε προηγουμένως. Ας υποθέσουμε επίσης ότι τον κυριεύει ο σατανάς...

```
*form_usr = xoxo'; DROP TABLE table_name; --
```

```
*form_pwd =
```

Συμπληρώνοντας τα πεδία με τον τρόπο που φαίνεται παραπάνω, η μεταβλητή που διατηρεί το ερώτημα προς τη βάση θα πάρει την ακόλουθη τιμή:

```
SELECT * FROM users WHERE username = 'xoxo'; DROP TABLE table_name;
-- AND password = '';
```

Αυτή τη φορά σχηματίζονται δύο επιμέρους ερωτήματα, καθώς κι ένα σχόλιο. Προφανώς το σχόλιο θα αγνοηθεί από τη βάση, ενώ το πρώτο ερώτημα θα επιστρέψει τα δεδομένα του χρήστη xoxo. Αυτό μπορεί να μην ακούγεται καλό αλλά υπάρχει και κάτι πολύ χειρότερο. Η εκτέλεση του DROP TABLE έχει σαν αποτέλεσμα τη διαγραφή ενός πίνακα. Έτσι, η εκτέλεση του δεύτερου ερωτήματος θα έχει σαν συνέπεια τη διαγραφή του πίνακα table_name!

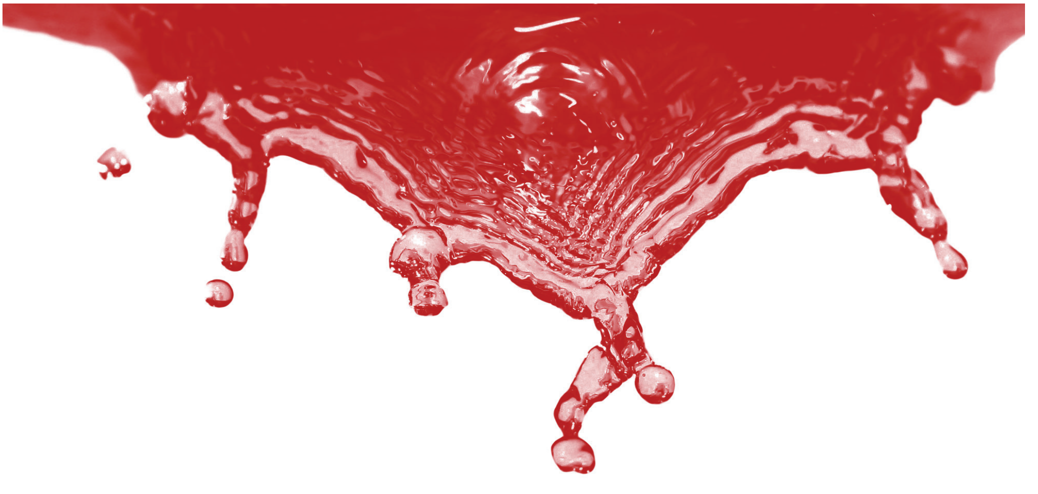
Συμμορφώστε τα δεδομένα

Η λογική που διέπει τις παραπάνω επιθέσεις είναι απλή κι αλάνθαστη. Βέβαια, αν ήταν τόσο εύκολο να συμβούν όσα περιγράψαμε, το Internet θα είχε παραδοθεί στο χάος: Ο καθένας θα μπορούσε να αποκτήσει πρόσβαση οπουδήποτε και να κάνει οτιδήποτε θελήσει. Όπως είπαμε νωρίτερα, ο αδύναμος κρίκος σε ένα site με δυναμικό περιεχόμενο είναι το πρόγραμμα που σχηματίζει τα ερωτήματα κι επικοινωνεί

με τη βάση δεδομένων. Ένα τέτοιο πρόγραμμα, όταν είναι καλογραμμένο ελέγχει τα δεδομένα που εισάγει ο χρήστης, τα τροποποιεί ή τα απορρίπτει μονομιάς και, τέλος πάντων, σε καμία περίπτωση δεν τα χρησιμοποιεί αυτούσια. Η διαδικασία που αναλαμβάνει αυτούς τους ελέγχους και πραγματοποιεί τις κατάλληλες παρεμβάσεις ονομάζεται sanitization. Επιγραμματικά, θα λέγαμε ότι το sanitization πραγματοποιεί κάποιες από τις ακόλουθες ενέργειες:

- Απομακρύνει τις λέξεις select, count, update, delete, drop καθώς και διάφορες άλλες, που ανήκουν στο λεξιλόγιο της SQL.
- Απομακρύνει τους χαρακτήρες που έχουν ιδιαίτερη/ειδική ερμηνεία στην SQL (π.χ. η απόστροφος, τα εισαγωγικά, οι παρενθέσεις, το ελληνικό ερωτηματικό κ.ά.).
- Προσθέτει το χαρακτήρα backslash (\) πριν από κάθε ειδικό χαρακτήρα. Αυτή η ενέργεια ονομάζεται escaping. Με τη βοήθεια του escaping, η SQL αντιμετωπίζει τους ειδικούς χαρακτήρες σαν όλους τους υπόλοιπους. Για παράδειγμα, οι δύο παύλες θα θεωρηθούν τμήμα κάποιου string και δεν θα αντιμετωπιστούν σαν την αρχή ενός σχολίου.
- Απορρίπτει πλήρως τα δεδομένα όταν τα θεωρεί επικίνδυνα ή ύποπτα.

Ελπίζουμε αυτή η πρώτη γνωριμία με τις επιθέσεις SQL injection να σας ανησύχησε λίγο ;) Φυσικά, σκοπός μας δεν είναι να σας τρομοκρατήσουμε, αλλά να επισημάνουμε την επικινδυνότητα αυτών των επιθέσεων, όπως επίσης και την τεράστια αξία που έχει η προσεκτική συγγραφή κώδικα. Αν παρακολουθείτε τη σειρά άρθρων για το LAMP, καλά θα κάνετε να είσαστε ιδιαίτερα προσεκτικοί με τις εφαρμογές που αναπτύσσετε και, όταν πρόκειται για εφαρμογές που αλληλεπιδρούν με το χρήστη, να μην εμπιστεύεστε κανέναν και τίποτα. Υποσχόμαστε να επανέλθουμε στο θέμα, με πιο ρεαλιστικά παραδείγματα και περισσότερες λεπτομέρειες.



Skill: Beginner
Tags: NTP, time server, VM, DDoS, NTP pool



by GiannouG

Δικτυακές καμπάνες!

Τις Παλιές Καλές Εποχές™, τότε που οι άνθρωποι δεν έτρεχαν πίσω από το ρολόι, η "πηγή" για τη ρύθμιση των ρολογιών ήταν μία: Η καμπάνα της εκκλησίας. Η ακρίβεια που προσέφερε αυτή η μέθοδος ήταν αρκετή για τις τότε εφαρμογές. Έπρεπε να μαζευτούν όλοι στην πλατεία του χωριού, στις 3 το μεσημέρι; Η καμπάνα ήταν εκείνη που θύμιζε σε όλους την ώρα της συνάντησης. Θα ερχόταν στις 5 το λεωφορείο; Ησυχία να ακούσουμε πότε θα χτυπήσει η καμπάνα, να πάμε να προϋπαντήσουμε τα αγαπημένα μας πρόσωπα από την Αθήνα. Προφανώς, αυτές τις ιστορίες τις ακούμε μόνο από όσους τις έζησαν. Οι συσκευές μας, όμως, εξακολουθούν να τις βιώνουν ακόμα και σήμερα.

Τι σχέση έχουν οι καμπάνες του ρολογιού της εκκλησίας, με τους υπολογιστές; Γιατί μπορεί να μας ενδιαφέρει κάτι τέτοιο; Αλήθεια, έχετε αναρωτηθεί πώς καταφέρνουν οι υπολογιστές που συνδέονται στο Internet, να έχουν την ίδια ώρα, χωρίς να "χάνει" το ρολόι κανενός; Για να λυθούν αυτά τα προβλήματα —διότι περί προβλημάτων πρόκειται—, θα μπορούσαμε να είχαμε εγκαταστήσει κάποιο είδος καμπάνας στο Internet. Στην πραγματικότητα, όμως, αντί για καμπάνες χρησιμοποιούνται κάποιοι servers, πάνω στους οποίους συνδέονται τα συστήματα και λαμβάνουν τη σωστή ώρα. Μπορείτε να φαντάζεστε αυτούς τους κόμβους σαν "reference points" ή, τέλος πάντων, σαν σημεία αναφοράς για το χρόνο.

Το πρόβλημα του συγχρονισμού των ρολογιών είναι ένα από τα μεγαλύτερα προβλήματα στα κατανεμημένα υπολογιστικά συστήματα. Ευτυχώς, έχουν βρεθεί αρκετές τεχνικές για την αντιμετώπισή του κι έχουν ενσωματωθεί σε ένα σχετικό πρωτόκολλο. Πρόκειται για το Network Time Protocol (NTP), το οποίο αναλαμβάνει τη σωστή μετάδοση της ώρας μέσω δικτύου, καταφέρνοντας να αντισταθμίσει το network latency. Είναι ένα από τα πιο σημαντικά πρωτόκολλα που λειτουργούν στο Internet και γενικότερα στα δίκτυα[1] TCP/IP του πλανήτη[2] μας. Υπάρχουν πολλοί servers που μοιράζουν την ώρα —πρόκειται για τους λεγόμενους time servers— και μπορούμε να συνδεθούμε σ' αυτούς ελεύθερα. Εξάλλου, μεγάλες εταιρείες σαν τη Microsoft και την Apple διαθέτουν τους δικούς τους time servers, για τους χρήστες των λειτουργικών συστημάτων τους.

Ο δικός μας NTP server

Τι θα λέγατε να στήναμε τον δικό μας time server; Τώρα σας ακούμε να διαμαρτύρεστε: "Μια στιγμή! Αφού υπάρχουν διαθέσιμοι τόσοι πολλοί time servers, γιατί να ασχοληθούμε κι εμείς;" Σε κάτι τέτοιες ερωτήσεις —και κυρίως όταν νιώθουμε λίγο στριμωγμένοι— μάς αρέσει ν' απαντάμε με ένα απλό "γιατί μπορούμε". Φυσικά, όπως με τα περισσότερα εγχειρήματά μας, αποσκοπούμε στο να γνωρίσουμε μία ακόμα τεχνολογία και, τέλος πάντων, ν' αποκτήσουμε νέες γνώσεις. Δεν σας κρύβουμε όμως ότι πολλές φορές το μοναδικό μας κίνητρο είναι απλά η διάθεση να περάσουμε ένα ενδιαφέρον σαββατιάτικο απόγευμα :D

Κουρδίζοντας τα ρολόγια

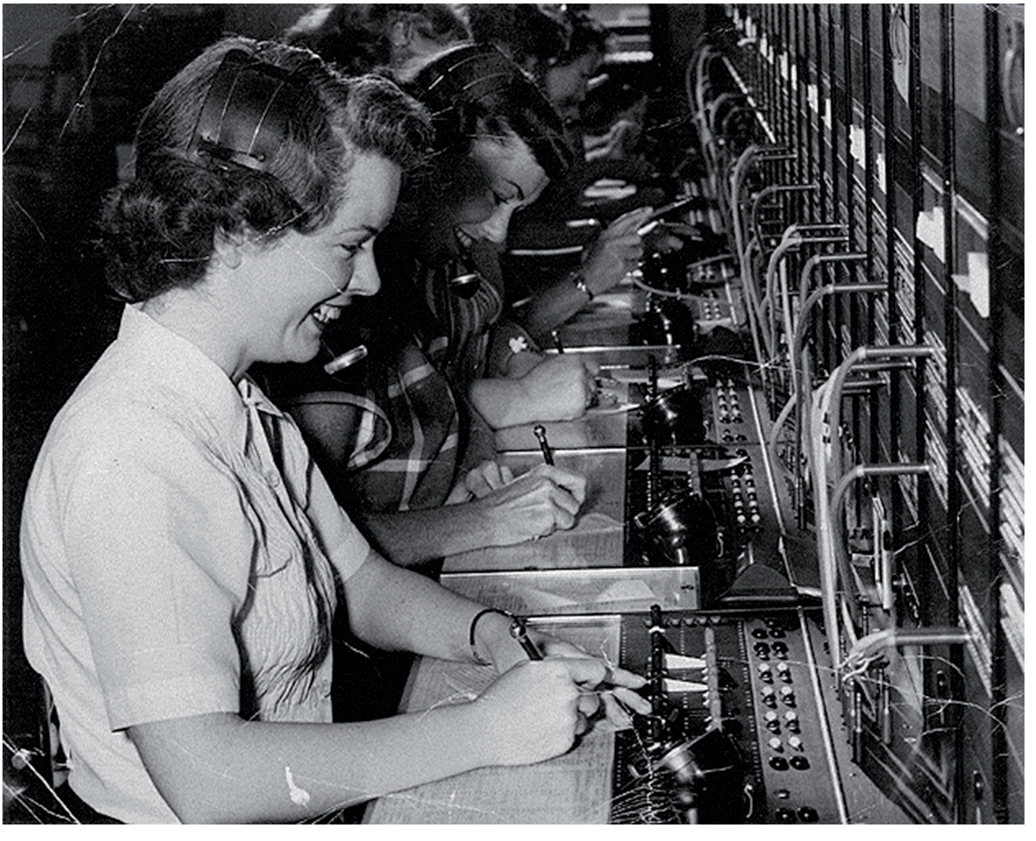
Τι θα χρειαστούμε άραγε για να στήσουμε τον δικό μας NTP server; Πριν απαντήσουμε σε αυτό το ερώτημα, πρέπει ν' απαντήσουμε σε κάποια άλλα: Θέλουμε να είναι διαθέσιμος σε όλο το Internet ή μόνο στα δικά μας μηχανήματα; Μήπως θέλουμε απλά να παίξουμε για μερικές ώρες το βροχερό Σαββατοκύριακο που έρχεται; Ποιο λειτουργικό είναι το καλύτερο για μια τέτοια δουλειά; Να βάλω Linux ή κάποιο *BSD; Μήπως να βάλω Windows, που τα ξέρω και καλύτερα;

Δε θα μακρηγορήσουμε άλλο. Όποιο λειτουργικό κι αν διαλέξουμε, ό,τι μηχάνημα κι αν διαθέτουμε, θα είναι κατάλληλο για τα περισσότερα σενάρια. Προφανώς, ένας δημόσιος NTP server που στηρίζεται στο Raspberry Pi θα ζοριστεί, θα γονατίσει και εκραγεί αφήνοντας ένα μανιτάρι που θα θυμίζει παλιό, ηλεκτρονικό παιχνίδι. Εμείς πάντως κάναμε τις δοκιμές μας πάνω σε ένα virtual machine με το Ubuntu 13.10.

Σε αυτό το σημείο θα ήταν καλό να αναφερθούμε σε ένα "μέτρο" ονόματι stratum, που χρησιμοποιείται στο πρωτόκολλο NTP. Για να καταλάβετε ευκολότερα τι είναι

1. AWMN κανείς; :D
2. Αν σκεφτούμε πως και ο Διεθνής Διαστημικός Σταθμός έχει Internet, τότε αυτή η φράση είναι λανθασμένη :P

αυτό το φρούτο, υποθέστε ότι έχουμε τρεις NTP servers. Ο τρίτος χρησιμοποιεί σαν πηγή συγχρονισμού τον δεύτερο κι αυτός με τη σειρά του τον πρώτο. Ο πρώτος time server επιστρατεύει ειδικό hardware και σε αυτό το ρόλο ενδέχεται να χρησιμοποιείται κάποιος δέκτης GPS ή κάποιο εξωτερικό κύκλωμα, το οποίο παράγει παλμούς χρονισμού (Pulse Per Second ή PPM). Το ειδικό hardware για την "παραγωγή" του χρόνου θεωρείται ως μια σταθερή και αξιόπιστη πηγή χρονισμού. Τα stratum δηλώνουν την "απόσταση" ενός time server από μία τέτοια πηγή. Έτσι, ο πρώτος από τους server που αναφέραμε στο παράδειγμα, λέμε ότι έχει stratum 1. Ο δεύτερος time server, που μαθαίνει την ώρα από τον πρώτο, έχει stratum 2 και ο τρίτος —σωστά μαντέψατε— έχει stratum 3. Οι servers με stratum 2 συνδέονται σε πολλούς servers με stratum 1, καθώς και σε άλλους του ίδιου "επιπέδου". Έτσι καταφέρνουν να διορθώσουν μικροπροβλήματα στο συγχρονισμό τους και να παρέχουν την ακριβέστερη δυνατή ώρα για τους servers με stratum 3. Κάτι παρόμοιο κάνουν και οι τελευταίοι, οι οποίοι συνδέονται σε πολλούς servers του ίδιου επιπέδου, όπως και σε αρκετούς με stratum 2. Η μέγιστη τιμή για το stratum είναι το 15. Το stratum 16 αποδίδεται μόνο στα μηχανήματα που θεωρούνται αποσυντονισμένα!

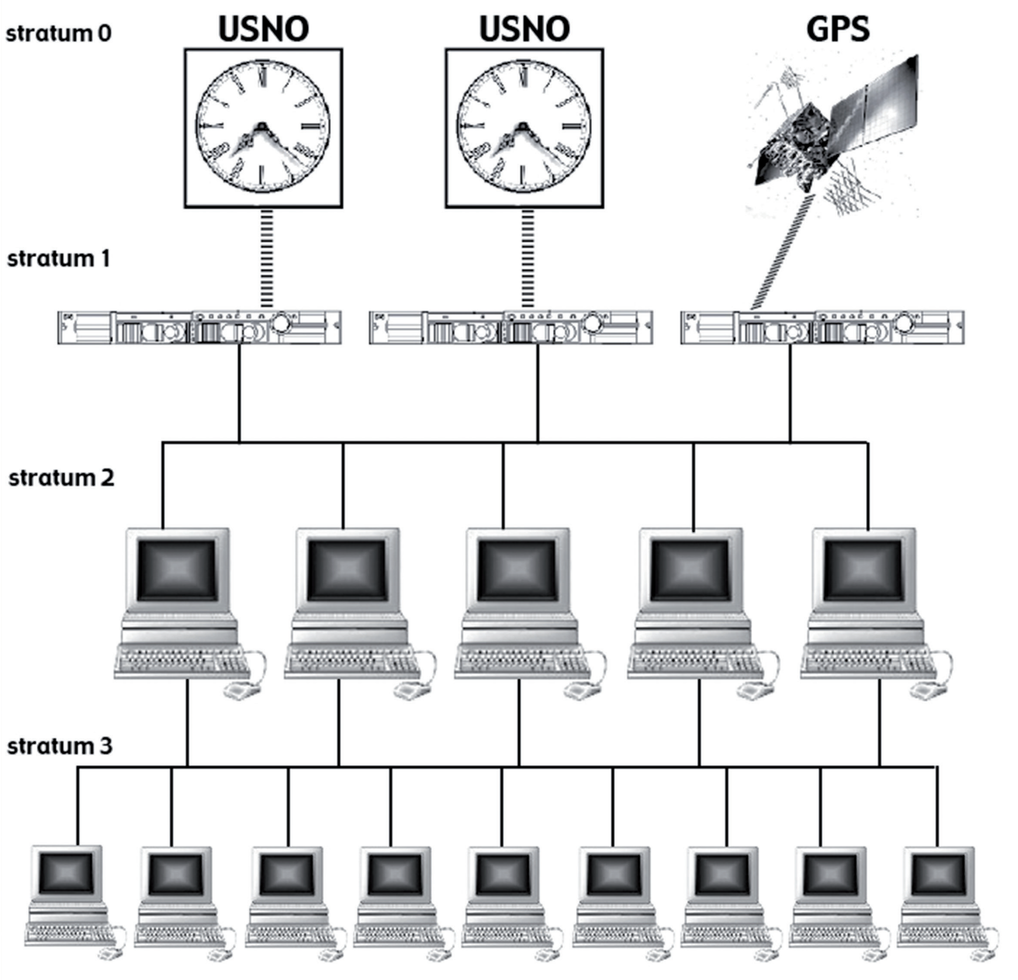


Οι εποχές έχουν αλλάξει αρκετά. Πλέον δεν χρειαζόμαστε ούτε την καμπάνα της εκκλησίας, ούτε το 141 για να συγχρονίσουμε τα ρολόγια μας.

Εγκατάσταση και ρύθμιση

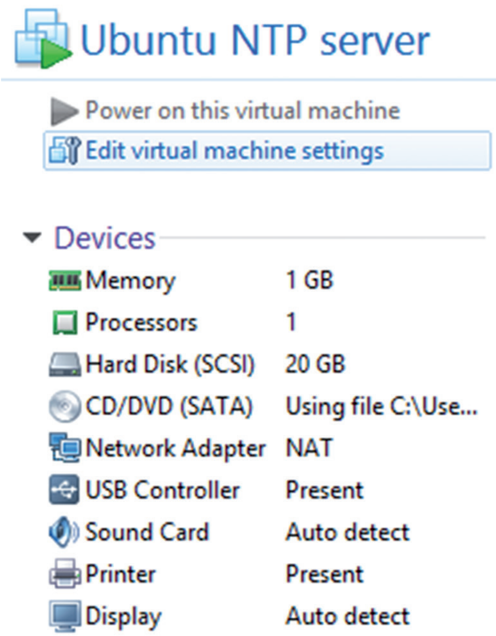
Από πλευράς software, αυτό που χρειαζόμαστε είναι το NTPd. Το NTP daemon —ή αλλιώς ο NTP server— τρέχει στο μηχάνημά μας και σε τακτά χρονικά διαστήματα επικοινωνεί με άλλους servers, ώστε να διατηρεί το ρολόι του όσο πιο καλά συγχρονισμένο γίνεται. Όπως καταλάβατε, δεν σκοπεύουμε να φτιάξουμε έναν time server με stratum 1. Κάτι τέτοιο θα απαιτούσε πανάκριβο εξοπλισμό που δεν διαθέτουμε[3]. Σκοπεύουμε όμως να δημιουργήσουμε έναν server με stratum 2! Έτσι, αφού βεβαιωθούμε ότι τα repositories του συστήματός μας είναι φρεσκαρισμένα, μπορούμε να προχωρήσουμε στην εγκατάσταση του NTP daemon:

```
sudo apt-get install ntp
```



Οι time server που "χρονίζονται" από GPS βρίσκονται στην κορυφή της ιεραρχίας. Τα μηχανήματα που "κρέμονται" κάτω από αυτά καταλαμβάνουν κατώτερες θέσεις στην ιεραρχία, σύμφωνα με την "απόστασή" τους.

3. Τουλάχιστον όχι ακόμα... :D



Ένα VM αποτελεί το ιδανικό σύστημα για το πειραματικό στήσιμο ενός NTP server ;)

Εκτός από τον NTPd, η εκτέλεση του παραπάνω θα προκαλέσει και την εγκατάσταση κάποιων συνοδευτικών προγραμμάτων, για τον έλεγχο της λειτουργίας του time server. Όταν ολοκληρωθεί η εγκατάσταση, μπορούμε να ξεκινήσουμε τη ρύθμιση του νέου μας server. Ανάλογα με την διανομή Linux που χρησιμοποιεί κανείς, τα σχετικά αρχεία ρυθμίσεων ενδέχεται να βρίσκονται σε διαφορετικές θέσεις. Στο Ubuntu, πάντως, και γενικότερα στις διανομές που βασίζονται στο Debian, το ζητούμενο αρχείο (ntp.conf) βρίσκεται χύμα στο /etc. Το ανοίγουμε με το αγαπημένο μας nano και ξεκινάμε.

```
sudo nano /etc/ntp.conf
```

Αυτό που θα παρατηρήσει κανείς ξεκινώντας αυτή την εργασία, είναι η *απίστευτα* απλή δομή του αρχείου ρυθμίσεων. Είναι σίγουρο ότι θα έχετε πραγματοποιήσει πολύ πιο σύνθετες ρυθμίσεις κι εδώ τα πράγματα θα σας φανούν εξαιρετικά απλά. Υπάρχουν μόνο δύο σημεία που απαιτούν λίγη παραπάνω προσοχή.

Στην αρχή του ntp.conf θα δούμε μερικές γραμμές που αναφέρονται στους servers, στους οποίους θα προσπαθεί να συνδέεται το μηχάνημά μας για το συγχρονισμό. Στο Ubuntu, οι servers αυτοί είναι μέλη του NTP pool project και έχουν τα hostnames `0.ubuntu.pool.ntp.org`, `1.ubuntu.pool.ntp.org`, `2.ubuntu.pool.ntp.org` και `3.ubuntu.pool.ntp.org`. Διαφορετικές διανομές ενδέχεται να χρησιμοποιούν διαφορετικά hostnames (Σ.τ.Ε. έτσι είναι), αλλά να είσαστε βέβαιοι ότι από πίσω βρίσκονται οι servers του NTP pool project. Εδώ θα πρέπει να ορίσουμε κάποιον server με stratum 1, ώστε ο δικός μας να έχει stratum 2! Αυτό δεν είναι αναγκαίο και θα μπορούσαμε να αφήσουμε αυτούς που προτείνει η διανομή μας, αλλά αν είσαστε κι εσείς τελειομανείς μπορείτε να καθορίσετε συγκεκριμένους time servers "χειροκίνητα". Ο ορισμός των servers γίνεται χρησιμοποιώντας τη λέξη "server" ακολουθούμενη από την IP ή το hostname του server.

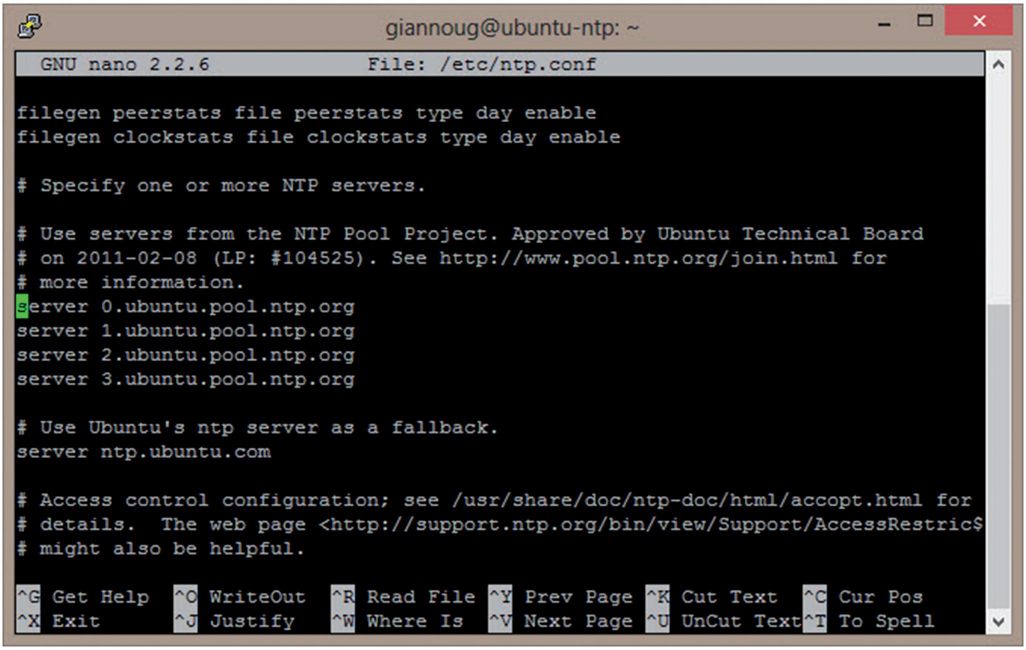
Μπορείτε να βρείτε πολλούς time servers με stratum 1 στο Google ή, πιο απλά, να ρίξετε μια ματιά στο `www.timetoolsglobal.com/2013/10/31/ntp-server-germany`, όπου αναφέρονται μερικοί "γειτονικοί" servers (βρίσκονται στη Γερμανία).

Το επόμενο τμήμα του αρχείου ρυθμίσεων στο οποίο ενδέχεται να θελήσουμε να επέμβουμε, είναι αυτό με τους περιορισμούς (οδηγία "restrict"). Εδώ πραγματοποιείται ο έλεγχος των δικαιωμάτων που έχουν οι χρήστες του NTP server. Οι προεπιλεγμένες ρυθμίσεις είναι οι πιο ασφαλείς και είναι κατάλληλες για τις περισσότερες περιπτώσεις. Όπως θα προσέξατε, ο NTP δεν αναρωτιέται για το interface του μηχανήματος στο οποίο θα ακούει, αφού από προεπιλογή ακούει σε όλα! Ωστόσο, μέσω του ορισμού "restrict" μπορούμε να κόψουμε διάφορες έξτρα λειτουργίες όπως, π.χ.,

η απομακρυσμένη ρύθμιση και κυρίως ορισμένες εντολές που μπορεί να δεχτεί ο server και οι οποίες θα απέβαιναν μοιραίες για τη σωστή λειτουργία του.

Αμέσως μετά τη σύντομη επέμβαση στο αρχείο ρυθμίσεων, μπορούμε να επανεκκινήσουμε τον NTP daemon. Αυτό μπορεί να γίνει πανεύκολα, με την εντολή service:

```
sudo service ntp restart
```



Μπορεί ο NTP daemon να είναι αρκετά απλός στο σετάρισμα, αλλά αυτό δεν σημαίνει ότι είναι και εντελώς ακίνδυνος. Όταν ρυθμίζουμε κάποια υπηρεσία, οφείλουμε να είμαστε πάντα προσεκτικοί!

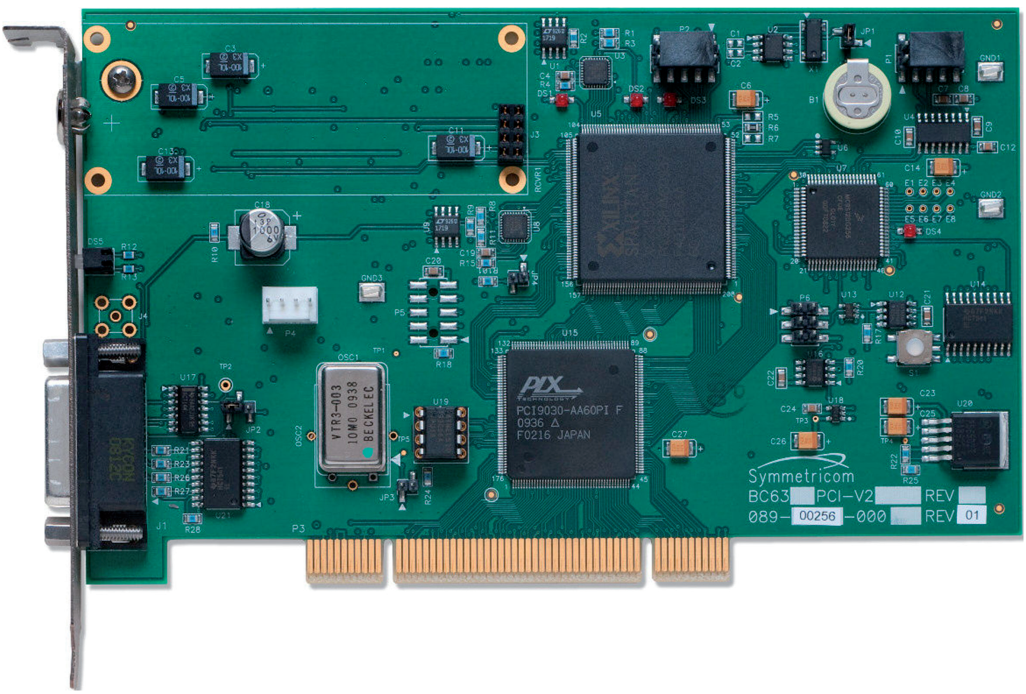
Μπελάδες απ' το πουθενά...

Χα! Σιγά το πράγμα! Μπορεί το πρωτόκολλο NTP να είναι ιδιαίτερα εξελιγμένο και να ενσωματώνει πολλές έξυπνες τεχνικές για τη σωστή μετάδοση της ώρας, αλλά το στήσιμο της σχετικής υπηρεσίας ήταν πανεύκολο. Τι μπορεί να πάει στραβά;

Στα τέλη του προηγούμενου έτους και μέσα στις πρώτες μέρες του νέου χρόνου, πραγματοποιήθηκαν μερικές ενδιαφέρουσες επιθέσεις σε δικτυακούς τόπους αλλά και σε server παιχνιδιών. Αναρωτιέστε τι κοινό είχαν οι επιθέσεις στο Origin της EA, στο Steam της Valve, στο Battle.net της Blizzard αλλά και στο League of Legends της Riot; Εκτός του ότι έγιναν σχεδόν ταυτόχρονα, βασίζονταν όλες σε μια παράβλεψη στη ρύθμιση του NTPd! Βλέπετε, οι αγαθούληδες (επιτρέψτε μας την έκφραση) προγραμματιστές του NTPd, είχαν υλοποιήσει μια εντολή που επέστρεφε μια λίστα με τους τελευταίους 600 πελάτες που συνδέθηκαν στο server και ζήτησαν την ώρα. Πρόκειται για την εντολή MONLIST, η οποία μπορεί να φανεί χρήσιμη σε όσους διατηρούν έναν NTP server. Ωστόσο, όπως πολύ καλά αντιλαμβάνεστε, η ύπαρξή της συνιστά ένα αρκετά μεγάλο κενό ασφαλείας. Ας δούμε το πρόβλημα με νούμερα. Στέλνουμε την εντολή αυτή σε κάποιον NTP server που ξέρουμε ότι θα απαντή-

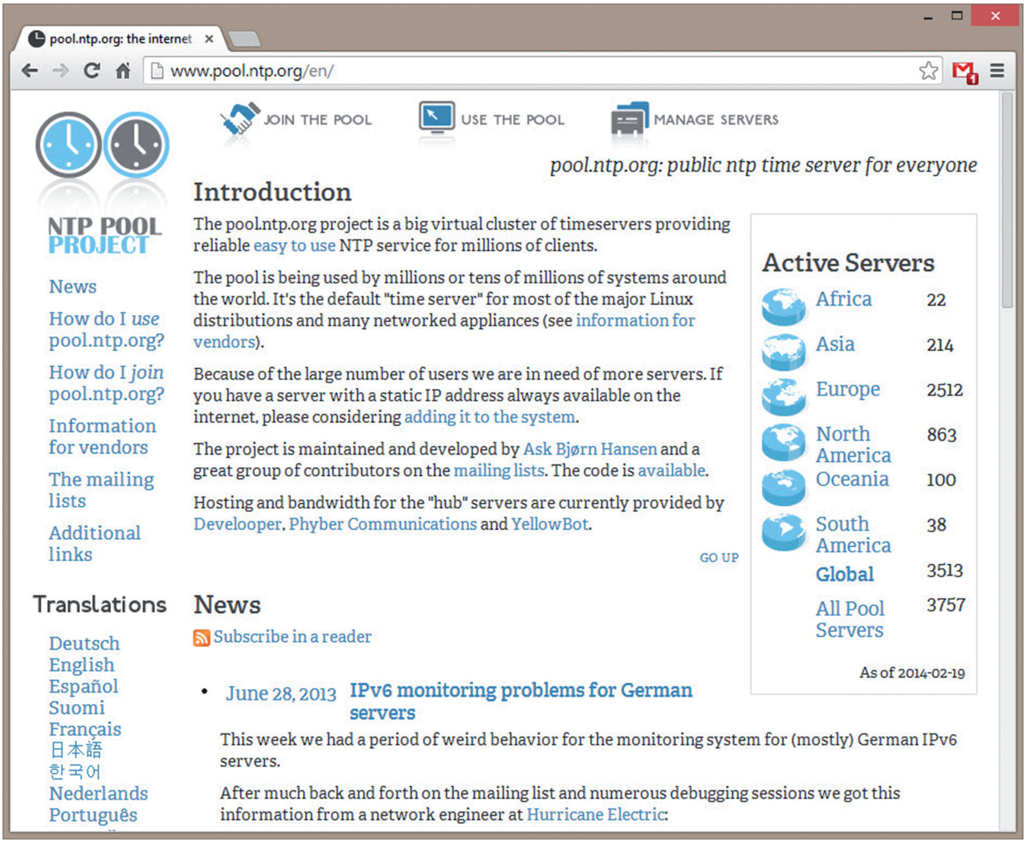


Στις περισσότερες των περιπτώσεων, οι εταιρείες χρησιμοποιούν ειδικό hardware για τον συγχρονισμό των ρολογιών τους. Οι συσκευές αυτές περιλαμβάνουν έναν δέκτη GPS, από τον οποίο λαμβάνουν την ώρα.



Σε περίπτωση που το απλό (!) GPS δεν είναι αρκετό, υπάρχει και η λύση των ατομικών ρολογιών ;) Θα χρειαστούμε κάτι σαν την πλακέτα της φωτογραφίας, που απλά γεφυρώνει τον υπολογιστή με κάποιο ατομικό ρολόι. Αναρωτιέστε για το κόστος; Φτάνει μόλις στα 500 ευρώ, αλλά μιλάμε για την πλακέτα. Όχι για το ατομικό ρολόι.

σει. Το UDP πακέτο που στέλνουμε "ζυγίζει" περί τα 250 byte. Η απάντηση που θα πάρουμε, ανάλογα του πόσα άτομα έχουν συνδεθεί, μπορεί να φτάσει μέχρι και τα 50KB! Αυτό σημαίνει ότι ο server στέλνει περί τα 250% περισσότερα δεδομένα, από αυτά που του στείλαμε. Τι θα γίνει αν καταφέρουμε να στείλουμε το σχετικό request, τροποποιώντας κατάλληλα το πακέτο UDP, ώστε στη θέση της δικής μας διεύθυνσης να φαίνεται μια άλλη; Ο NTP server, άθελά του, θα αρχίσει να απαντάει



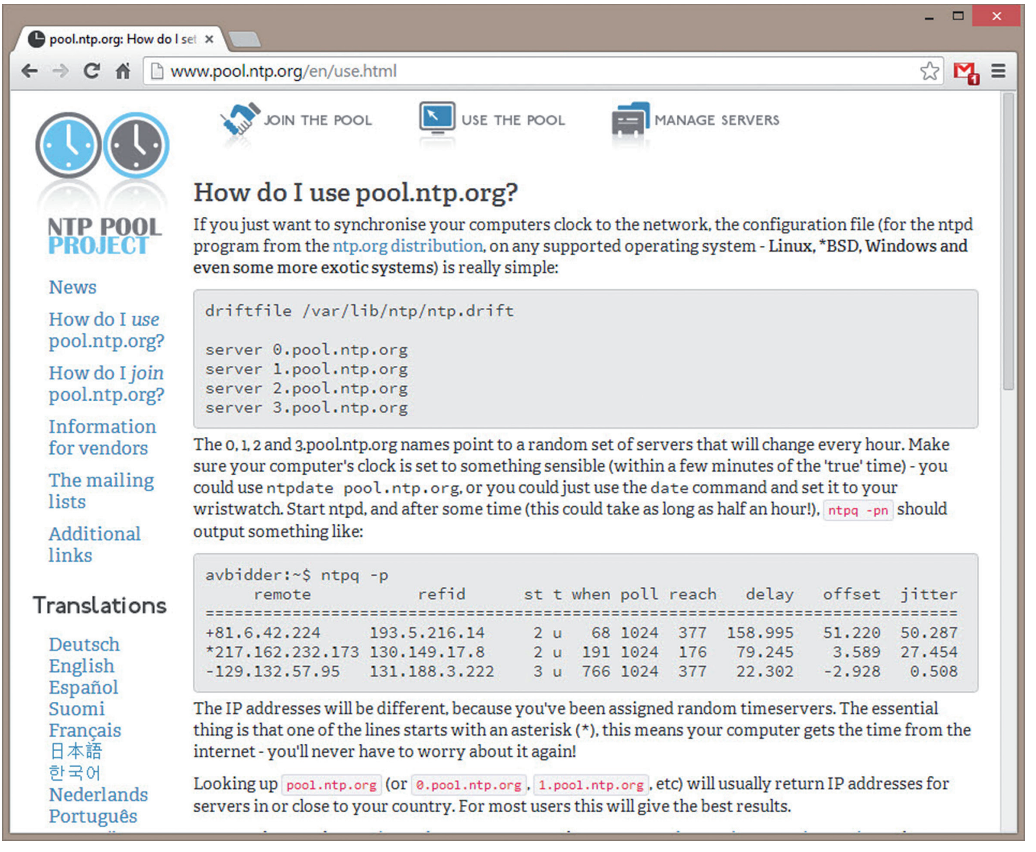
Το NTP pool project αποτελεί τη μεγαλύτερη λίστα δημόσιων NTP server. Οι servers είναι χωρισμένοι σε περιοχές, ανάλογα με την ήπειρο και τη χώρα στην οποία βρίσκεται ο καθένας.

σε λάθος μηχάνημα! Φανταστείτε τώρα πολλούς NTP servers να στέλνουν τη λίστα με τους πελάτες τους σε μηχανήματα που ποτέ δεν ζήτησαν κάτι τέτοιο: Μιλάμε για εύκολο και δωρεάν DDoS!

Δεν υπάρχει λόγος για πανικό. Οι περισσότεροι NTP servers είναι *σωστά* ρυθμισμένοι εξ αρχής και η εν λόγω εντολή γίνεται δεκτή μόνον όταν προέρχεται από το ίδιο μηχάνημα. Αυτό που πρέπει να κρατήσουμε, είναι ότι και στην περίπτωση μιας απλής κι αθώας υπηρεσίας υπάρχει πάντα η πιθανότητα να πάει κάτι στραβά. Πολύ στραβά! Άντε μετά να αποδείξεις ότι δεν ευθύνεσαι εσύ για το γεγονός ότι ο server σου έγινε συμμέτοχος σε μία επίθεση DDoS.

Ωραιότατος NTP server

Αφού είδαμε το ομολογουμένως πανεύκολο σετάρισμα ενός NTP server και μετά την περιγραφή ενός κινδύνου που κρύβει η αφελής ρύθμισή του, έφτασε νομίζουμε η ώρα να δούμε πώς θα τον χρησιμοποιήσουμε! Θα εξετάσουμε δύο περιπτώσεις: Μια για τα δικά μας μηχανήματα και μια για όλα τα υπόλοιπα συστήματα του κόσμου τούτου, μέσω του NTP pool project.



Η χρήση του NTP pool είναι αρκετά απλή. Αρκεί να χρησιμοποιού-
με το hostname κάποιου από τους διαθέσιμους time servers.

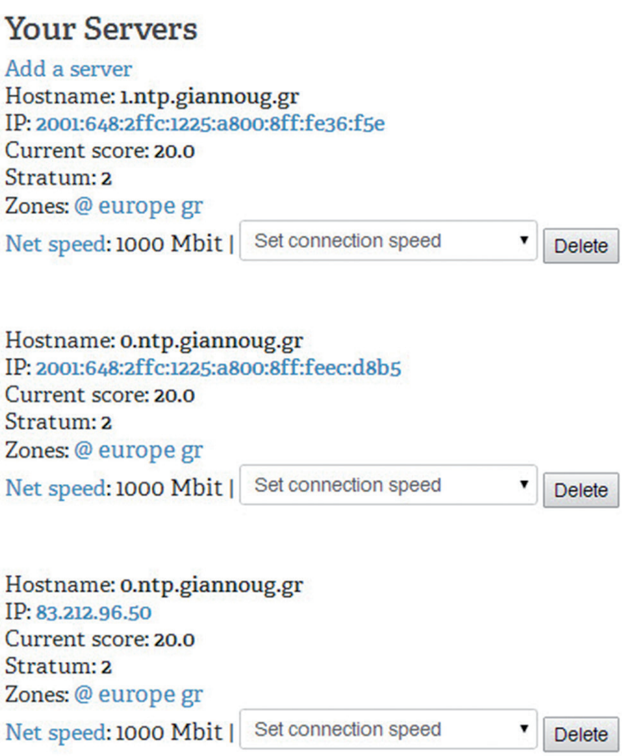
Προκειμένου να χρησιμοποιήσουμε το νέο μας NTP server σε κάποιο σύστημα Windows, αρκεί να μεταβούμε στην καρτέλα "Internet Time", στο παραθυράκι ρύθμισης της ώρας. Πατώντας "Change settings" θα εμφανιστεί ένα πλαίσιο για τον ορισμό ενός time server. Εκεί αρκεί να δώσουμε τη διεύθυνση IP του μηχανήματός μας. Πατώντας "Update now", ο υπολογιστής μας θα συγχρονιστεί και το ρολόι του συστήματός μας θα πάρει τη "φρέσκια" ώρα που μοιράζει ο server μας με stratum 2!

Ωραία, τώρα μπορούμε να τον απολαύσουμε, αλλά *μόνοι* μας. Δεν θα νιώθατε καλύτερα αν μοιράζατε τις υπηρεσίες του και στο υπόλοιπο Internet; Για αυτό το σκοπό υπάρχει το NTP pool project (`www.pool.ntp.org`), το οποίο συγκεντρώνει και διαχειρίζεται όλους τους δημόσιους και δωρεάν NTP servers! Για να γίνει κάποιος μέλος, θα πρέπει να έχει ένα αξιοπρεπές μηχάνημα καθώς και μια σταθερή διεύθυνση IP στο Internet. Αφού βεβαιωθούμε ότι ο server μας λειτουργεί σωστά, το μόνο που χρειάζεται να κάνουμε είναι να γραφτούμε στην υπηρεσία και να προσθέσουμε τη δημόσια IP του μηχανήματος. Η υπηρεσία θα αναγνωρίσει σε ποια χώρα βρίσκεται ο server μας και θα την προσθέσει στην ανάλογη ζώνη. Ανάλογα με το πόσο καλά διατηρεί το χρόνο ο server μας, θα πάρει και μια βαθμολογία. Από αυτό το σκορ καθορίζεται αν το μηχάνημά μας θα παραμείνει μέσα στο pool ή όχι. Όπως αντιλαμβάνεστε, ο

πλέον καθοριστικός παράγοντας για μια καλή βαθμολογία, είναι η σωστή επιλογή των servers από τους οποίους θα συγχρονίζεται ο δικός μας ;)

Με την ευκαιρία, μιλώντας για τους servers από τους οποίους συγχρονίζεται ο δικός μας, αξίζει να αναφερθούμε στο εργαλείο ntpq (το όνομά του προκύπτει από το "NTP query"). Αν το εκτελέσουμε με την παράμετρο –p, εμφανίζει όλους τους servers που έχουμε επιλέξει από το αρχείο ρυθμίσεων, για το συγχρονισμό του δικού μας. Αν τρέξετε το πρόγραμμα με αυτή την παράμετρο και παρατηρήσετε το output, θα διαπιστώσετε ότι πριν από έναν server εμφανίζεται ένας αστερίσκος. Η παρουσία του δηλώνει ότι ο συγκεκριμένος server είναι αυτός που επιλέχθηκε από το σύστημά μας για το συγχρονισμό. Στην ίδια γραμμή και στη στήλη poll, θα δούμε κι έναν αριθμό. Πρόκειται για το χρονικό διάστημα ανά το οποίο ενημερώνεται το ρολόι του server μας. Ένας τρόπος για να βελτιώσουμε τη βαθμολογία που δίνει το NTP pool project στο μηχάνημά μας, είναι να μικρύνουμε αυτό ακριβώς το μέγεθος. Η σχετική επέμβαση γίνεται από το αρχείο ρυθμίσεων του NTP daemon και είμαστε σίγουροι ότι θα την εντοπίσετε πανεύκολα. Προσοχή, όμως! Μην το παρακάνετε επιλέγοντας ένα εξαιρετικά μικρό χρόνο ανανέωσης, γιατί θα στουμπώσετε τη σύνδεσή σας.

Για την ιστορία, ο δικός μας server μας έχει stratum 2 και τις τελευταίες δύο εβδομάδες, το NTP pool project μάς δίνει βαθμολογία 20.



Οι δικοί μας servers έχουν το μέγιστο σκορ στο pool! Εδώ και μερικές βδομάδες απαντούν καθημερινά σε εκατοντάδες requests :D

Skill: Intermediate
Tags: LAMP, web, development, DDL, DML, MySQL



```
mysql> describe messages ;
+-------+------------------+------+-----+---------+----------------+
| Field | Type             | Null | Key | Default | Extra          |
+-------+------------------+------+-----+---------+----------------+
| id    | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name  | char(50)         | NO   |     | NULL    |                |
| user  | int(10) unsigned | NO   |     | NULL    |                |
| msg   | text             | NO   |     | NULL    |                |
| date  | datetime         | NO   |     | NULL    |                |
+-------+------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> alter table messages drop name ;
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe messages ;
+-------+------------------+------+-----+---------+----------------+
| Field | Type             | Null | Key | Default | Extra          |
+-------+------------------+------+-----+---------+----------------+
| id    | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| user  | int(10) unsigned | NO   |     | NULL    |                |
| msg   | text             | NO   |     | NULL    |                |
| date  | datetime         | NO   |     | NULL    |                |
+-------+------------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

by multiPetros

Web development σε περιβάλλον LAMP [Μέρος 6]

Μέχρι στιγμής έχουμε μάθει τα βασικά για τη δημιουργία και τη διαχείριση πινάκων και βάσεων στη MySQL. Όμως τι νόημα έχει μια βάση δεδομένων χωρίς δεδομένα; Σε αυτό το άρθρο θα αφήσουμε πίσω μας το τμήμα της γλώσσας που αφορά στον ορισμό βάσεων, πινάκων και πεδίων, για να προχωρήσουμε στο τμήμα της γλώσσας που σχετίζεται με την προσθήκη και το χειρισμό του περιεχομένου...

Στο προηγούμενο τεύχος γνωρίσαμε τις βασικές έννοιες των βάσεων δεδομένων και αποκτήσαμε μια πρώτη επαφή με τη γλώσσα της MySQL. Πλέον, είμαστε σε θέση να δημιουργήσουμε μια βάση δεδομένων και να προσθέσουμε πίνακες με οσεσδήποτε γραμμές και στήλες. Όλα αυτά βέβαια αποτελούν τις απολύτως βασικές γνώσεις που μπορεί να έχει κανείς γύρω από το θέμα. Όπως καταλαβαίνετε, τα ίδια "όλα αυτά" δεν θα μπορούσαν να είναι αρκετά για εμάς! Σε αυτό το μέρος θα γνωρίσουμε τα εργαλεία"που προσφέρει η γλώσσα της MySQL για τη διαχείριση του περιεχομένου. Παρεμπιπτόντως, να πούμε ότι η MySQL[1] χρησιμοποιεί μια δική της παραλλαγή της τυποποιημένης ANSI SQL και συγκεκριμένα ένα υπερσύνολό της. Το κομμάτι της γλώσσας που αφορά στη δομή των βάσεων και των πινάκων ονομάζεται DDL (Data Definition Language), ενώ το κομμάτι που αφορά στη διαχείριση των δεδομένων ονομάζεται DML (Data Manipulation Language).

Εισαγωγή δεδομένων

Θα δουλέψουμε στον πίνακα messages της βάσης δεδομένων gbook, που δημιουργήσαμε στο προηγούμενο άρθρο. Για να ξεκινήσουμε, συνδεόμαστε στην κονσόλα του προσφιλούς, πλέον, MySQL Monitor. Αφού δηλώσουμε ότι σκοπεύουμε να εργαστούμε στη βάση δεδομένων gbook, χρησιμοποιούμε ένα ερώτημα insert για να εισαγάγουμε μια εγγραφή στον πίνακα:

```
use gbook ;

insert into messages values (NULL, 'deltaFun', 'keep going!', now()) ;
```

Αυτό ήταν! Μόλις προσθέσαμε μια γραμμή στον πίνακα messages. Οι τιμές που δώσαμε μέσα στην παρένθεση αντιστοιχούν στις τιμές των πεδίων που απαρτίζουν τη γραμμή. Όπως θα παρατηρήσατε, για το πρώτο πεδίο (αναγνωριστικός κωδικός της εγγραφής) δεν δώσαμε τιμή (NULL). Αυτό το κάναμε γιατί έχουμε επιλέξει την αυτόματη συμπλήρωση του συγκεκριμένου πεδίου από τη MySQL[2]. Σημειώστε ότι ενώ οι συμβολοσειρές πρέπει να περικλείονται πάντα από εισαγωγικά, οι αριθμητικές τιμές (άρα και οι ημερομηνίες) αναφέρονται... χύμα. Τέλος, αν αναρωτιέστε τι είναι αυτό το περίεργο now(), πρέπει να σας πούμε ότι πρόκειται για μια συνάρτηση. Βλέπετε, η MySQL υποστηρίζει τη χρήση συναρτήσεων, διαθέτει αρκετές ενσωματωμένες κι επιτρέπει στο χρήστη να ορίζει και τις δικές του. Όλα αυτά όμως δεν θα μας απασχολήσουν επί του παρόντος. Όπως καταλαβαίνετε, η συνάρτηση now() επιστρέφει την τρέχουσα ημερομηνία.

Η εντολή insert υποστηρίζει και μια εναλλακτική σύνταξη, με την οποία μπορούμε να αποδώσουμε τιμές *μόνο* σε όσα πεδία θέλουμε. Δείτε ένα παράδειγμα:

```
insert into messages (name, msg, date) values ('deltaFun', 'keep going!',
now()) ;
```

Με αυτόν τον τρόπο προσθέτουμε μια νέα εγγραφή, για την οποία ορίζουμε μόνο τα πεδία name, msg και date. Μπορούμε να πετύχουμε το ίδιο αποτέλεσμα και μ' έναν ακόμα τρόπο:

```
insert into messages name='deltaFun', msg='keep going!', date=now() ;
```

1. Το πρώτο συνθετικό του ονόματος της "MySQL" δεν προέρχεται από την αντωνυμία my, αλλά από το όνομα της πρωτότοκης κόρης του Monty Widenius, βασικού δημιουργού της MySQL. Ο ίδιος άνθρωπος βρίσκεται πίσω και από την πολλά υποσχόμενη MariaDB, η οποία πήρε το όνομά της από το όνομα της 2ης κόρης του, σε μια προσπάθεια να ξεπεραστούν τα προβλήματα στην ανάπτυξη της MySQL (και άλλων δημοφιλών projects) που δημιούργησε η εξαγορά της Sun από την Oracle.

2. Όπως θα δείτε και στη σελίδα 66 του περασμένου τεύχους (028), όταν ορίσαμε το πρώτο πεδίο (id) του δώσαμε την ιδιότητα auto_increment.

Επιπρόσθετα, συντάσσοντας κατάλληλα το ερώτημα insert, μπορούμε να εισάγουμε ταυτόχρονα πολλές εγγραφές. Για να πετύχουμε κάτι τέτοιο, αρκεί να χρησιμοποιήσουμε ένα ζεύγος παρενθέσεων για κάθε νέα εγγραφή και να χωρίσουμε τις εγγραφές με κόμματα...

```
insert into messages values
(NULL, 'barba ben', 'na pate allou kyrioi', now()),
(NULL, 'lame', 'geniki apergia, giati theloume piso to palio twitter...',
now()),
(NULL, 'lucky reader', 'euxaristo gia ton SSD! :)', now()) ;
```

Ανάκληση δεδομένων

Όμορφα! Τώρα που προσθέσαμε μερικές εγγραφές στον πίνακα messages, ήρθε η ώρα να τις ανακαλέσουμε: Να τις διαβάσουμε από τον πίνακα και να τις προβάλλουμε. Γι' αυτή την εργασία θα χρησιμοποιήσουμε ένα ερώτημα select:

```
select * from messages ;
```

Εκτελώντας το παραπάνω, το MySQL Monitor θα εμφανίσει όλες τις εγγραφές του πίνακα messages (όλες τις γραμμές και όλες τις στήλες). Είναι επειδή ο αστερίσκος (*) λειτουργεί σαν μπαλαντέρ — στη θέση του θα μπορούσαμε να χρησιμοποιήσουμε πολλά άλλα στοιχεία. Για παράδειγμα, θα μπορούσαμε να αναφέρουμε μερικά ονόματα πεδίων. Εξετάστε το ακόλουθο παράδειγμα:

```
select msg, date from messages ;
```

Η εκτέλεση της παραπάνω γραμμής θα επιστρέψει όλες τις εγγραφές του πίνακα messages, αλλά σε καθεμία θα εμφανίζονται μόνο τα πεδία (οι στήλες) msg και date.

Αναζητήσεις

Η τεράστια δύναμη των ερωτημάτων select εντοπίζεται σε μια σημαντική δυνατότητα που προσφέρουν: αυτή της αναζήτησης! Με ένα ερώτημα select μπορούμε να προσδιορίσουμε το σύνολο των δεδομένων εντός των οποίων θέλουμε να εντοπίσουμε κάτι και στη συνέχεια να ορίσουμε τα επιθυμητά κριτήρια αναζήτησης. Τα κριτήρια, που μπορείτε να τα φαντάζεστε και σαν φίλτρα, εκφράζονται με τη βοήθεια διαφόρων τελεστών, που μπορείτε να δείτε στον ακόλουθο πίνακα:

Τελεστής	Περιγραφή	Χρήση
=	Έλεγχος ισότητας	name = 'deltaFun'
>	Μεγαλύτερο	date > now()
<	Μικρότερο	date < now()
>=	Μεγαλύτερο ή ίσο	date >= now()

Τελεστής	Περιγραφή	Χρήση
<=	Μικρότερο ή ίσο	date <= now()
!= ή <>	Διάφορο	id != 0 ή id <> 0
is null	Αν δεν έχει τιμή	msg is null
is not null	Αν δεν είναι	msg is not null
between	Μεταξύ δύο τιμών	id between 0 and 10
in	Εντός ενός συνόλου τιμών	name in ('lucky reader', 'deltaFun')
not in	Εκτός ενός συνόλου τιμών	name not in ('lame', 'barba ben')
like	Όμοιο μ' ένα μοτίβο	name like ('delta%')
not like	Ανόμοιο μ' ένα μοτίβο	name not like ('bar_ ben')
regexp ή rlike	Ταίριασμα με μια κανονική έκφραση[3]	id rlike '[1-5]'

Αξίζει να σημειώσουμε ότι μπορούμε να "επεκτείνουμε" μια αναζήτηση σε περισσότερους από έναν πίνακες. Αυτή η δυνατότητα πηγάζει από την ικανότητα της MySQL να "ενώνει" ξεχωριστούς πίνακες και αποτελεί χαρακτηριστικό γνώρισμα των σχεσιακών βάσεων δεδομένων. Η ένωση των πινάκων που αναφέραμε, μπορεί να εκφραστεί μαθηματικά με το γινόμενο πινάκων[4]. Συντακτικά, για να πραγματοποιήσουμε μια αναζήτηση εντός δύο ή περισσοτέρων πινάκων αρκεί να παραθέσουμε τα ονόματά τους, χωρίζοντάς τα με κόμματα. Όταν σχηματίζουμε ένα τέτοιο ερώτημα, πρέπει κάθε φορά που αναφερόμαστε σε ένα πεδίο να δηλώνουμε και σε ποιον πίνακα ανήκει. Αυτό γίνεται γράφοντας το όνομα του πίνακα, μια τελεία και στη συνέχεια το όνομα του επιθυμητού πεδίου. Κι επειδή αυτή η περιγραφή ενδέχεται να σας μπέρδεψε, ας δούμε κι ένα (διαφωτιστικό) παράδειγμα:

```
select pinakas_a.pedio_a, pinakas_b.pedio_c from pinakas_a, pinakas_b
where pinakas_a.pedio_a = 'something' ;
```

3. Μάθετε τα πάντα για τις κανονικές εκφράσεις, στα άρθρα του φίλου Spir@lEvolution, που φιλοξενούνται στα τεύχη 021 έως και 024 του deltaHacker!
4. Είναι λίγο πιο σύνθετη πράξη από αυτή που θα φανταζόταν κανείς ;-)

Το παραπάνω ερώτημα θα πραγματοποιήσει μια αναζήτηση στην ένωση των πινάκων pinakas_a και pinakas_b. Το κριτήριο της αναζήτησης εκφράζεται στο τέλος του ερωτήματος και προσδιορίζει αυτό που ζητάμε: Τις εγγραφές στις οποίες το πεδίο pedio_a του pinakas_a έχει την τιμή 'something'. Παρατηρώντας το αρχικό τμήμα του ερωτήματος, βλέπουμε ότι οι εγγραφές που ικανοποιούν το κριτήριό μας δεν θα εμφανιστούν ολόκληρες. Αντίθετα, θα εμφανιστούν μόνον οι στήλες (τα πεδία) pedio_a του pinakas_a και pedio_c του pinakas_b. Αν επιθυμούσαμε την εμφάνιση ολόκληρων των εγγραφών, θα αρκούσε να τοποθετήσουμε έναν αστερίσκο ανάμεσα στις λέξεις select και from.

Ταξινόμηση αποτελεσμάτων

Τα αποτελέσματα που λαμβάνουμε από ένα ερώτημα επιλογής εμφανίζονται με τη χρονολογική σειρά προσθήκης των αντίστοιχων εγγραφών. Αν θέλουμε να τα ταξινομήσουμε ως προς κάποιο πεδίο και με κάποια συγκεκριμένη σειρά (αύξουσα ή φθίνουσα), πρέπει να χρησιμοποιήσουμε την έκφραση "order by" στο τέλος του ερωτήματος. Για την ακρίβεια, μετά το "order by" πρέπει να αναφέρουμε το όνομα του πεδίου ως προς το οποίο θέλουμε να γίνει η ταξινόμηση και, προαιρετικά, να συνεχίσουμε καθορίζοντας την επιθυμητή σειρά. Εξ ορισμού η ταξινόμηση ακολουθεί αύξουσα σειρά, ενώ μπορούμε να την αντιστρέψουμε χρησιμοποιώντας τη λέξη desc. Στα παραδείγματα που ακολουθούν ζητάμε όλες τις εγγραφές του πίνακα messages, των οποίων το πεδίο id έχει τιμή μεγαλύτερη του 1. Στο πρώτο παράδειγμα, τα αποτελέσματα θα εμφανιστούν ταξινομημένα ως προς το πεδίο name και με αύξουσα σειρά. Στο δεύτερο, τα αποτελέσματα θα ταξινομηθούν με φθίνουσα σειρά:

```
select * from messages where id > 1 order by name ;
select * from messages where id > 1 order by name desc ;
```

Σύνοψη αποτελεσμάτων

Η MySQL προσφέρει ένα σύνολο εξειδικευμένων συναρτήσεων, με τη βοήθεια των οποίων μπορούμε να συνοψίσουμε τα αποτελέσματα μιας αναζήτησης. Ο ακόλουθος πίνακας περιλαμβάνει μερικές από αυτές τις συναρτήσεις, μαζί με μια σύντομη περιγραφή.

Συνάρτηση	Λειτουργία
avg(στήλη)	Μέσος όρος τιμών της στήλης
count(στήλη)	Καταμέτρηση συμπληρωμένων κελιών της στήλης
count(distinct στήλη)	Καταμέτρηση διαφορετικών κελιών της στήλης
count(*)	Καταμέτρηση όλων των κελιών του επιστρεφόμενου πίνακα

Συνάρτηση	Λειτουργία
max(στήλη)	Μέγιστη τιμή της στήλης
min(στήλη)	Ελάχιστη τιμή της στήλης
std(στήλη)	Τυπική απόκλιση των τιμών της στήλης
sum(στήλη)	Άθροισμα των τιμών της στήλης

Ας δούμε κι ένα παράδειγμα τώρα. Ας υποθέσουμε ότι θέλουμε να μάθουμε πόσες εγγραφές έχει ο πίνακας messages. Για να πετύχουμε κάτι τέτοιο, αρκεί να μετρήσουμε τα συμπληρωμένα κελιά της στήλης id. Φυσικά, θα μπορούσαμε να μετρήσουμε και τα συμπληρωμένα κελιά μιας άλλης στήλης. Ωστόσο είναι ασφαλέστερο να μετρήσουμε αυτά της στήλης id, καθώς η συγκεκριμένη στήλη συμπληρώνεται αυτόματα και καμία εγγραφή δεν μπορεί να έχει κενό αυτό το πεδίο. Δείτε το σχετικό ερώτημα:

```
select count(id) from messages ;
```

Ομαδοποίηση αποτελεσμάτων

Μια ακόμα εξυπηρετική λειτουργία είναι η ομαδοποίηση των δεδομένων. Αυτή η δυνατότητα, σε συνδυασμό με τη σύνοψη, επιταχύνει και διευκολύνει την εξαγωγή χρήσιμων συμπερασμάτων για τα περιεχόμενα ενός πίνακα. Η ομαδοποίηση πραγματοποιείται με τη χρήση της φράσης "group by". Σημειώστε τώρα ότι σε συνδυασμό με το "group by" μπορούμε να χρησιμοποιήσουμε και τον όρο "having" (παρόμοιος με τον "where"), ώστε μαζί με την ομαδοποίηση να πετύχουμε ταυτόχρονα και ένα φιλτράρισμα. Σε αυτό το σημείο είναι πολύ πιθανό να σας μπερδέψαμε. Ας δούμε και πάλι ένα παράδειγμα: Ας υποθέσουμε ότι θέλουμε να μάθουμε το πλήθος των μηνυμάτων που έχουν αποθηκευτεί στον πίνακα messages, αφού τα ομαδοποιήσουμε σύμφωνα με την ημερομηνία δημιουργίας κι αφού αποκλείσουμε τα μηνύματα που είναι παλιότερα από κάποια ημερομηνία. Αυτό το ερώτημα ακούγεται αρκετά περίπλοκο, αλλά δείτε πόσο εύκολα μπορεί να υλοποιηθεί:

```
select date, count(id) from messages group by date having date > '2014-01-
01' ;
```

Εκτελώντας την παραπάνω γραμμή, τα μηνύματα του πίνακα messages που δημιουργήθηκαν μετά τη 1/1/2014 θα ομαδοποιηθούν σύμφωνα με το χρόνο δημιουργίας τους και για κάθε ομάδα θα εμφανιστεί μόνο το πλήθος.

Περιορισμός αποτελεσμάτων

Μια χρήσιμη προσθήκη της MySQL, που ξεφεύγει από τα πρότυπα της "επίσημης" SQL, αποτελεί ο όρος limit. Όπως καταλαβαίνετε, με αυτόν τον όρο μπορούμε να περιορίσουμε το πλήθος των αποτελεσμάτων που λαμβάνουμε από την εκτέλεση ενός ερωτήματος επιλογής (select). Κάτι τέτοιο είναι ιδιαίτερα χρήσιμο στην ανάπτυξη

εφαρμογών web. Για παράδειγμα, σκεφτείτε την περίπτωση που θέλουμε τα *πέντε* πιο πρόσφατα posts μιας κάποιας κατηγορίας.

Ο όρος limit απαιτεί δύο παραμέτρους, των οποίων οι τιμές χωρίζονται με κόμμα. Η πρώτη καθορίζει την εγγραφή (γραμμή) του πίνακα από την οποία κι έπειτα θέλουμε να πάρουμε αποτελέσματα, ενώ η δεύτερη καθορίζει το πλήθος τους. Έστω λοιπόν ότι θέλουμε να πάρουμε δύο εγγραφές από τον πίνακά μας, ξεκινώντας από την δεύτερη εγγραφή:

```
select * from messages limit 1, 2
```

Δεν ξέρουμε αν σας έκανε εντύπωση, αλλά ενώ θέλουμε να ξεκινήσουμε από τη δεύτερη εγγραφή, η σχετική παράμετρος του όρου limit έχει την τιμή 1. Αυτό συμβαίνει γιατί η αρίθμηση των εγγραφών ξεκινά από το μηδέν.

Ενημέρωση δεδομένων

Μέχρι στιγμής έχουμε μάθει να εισάγουμε και να λαμβάνουμε δεδομένα, με τα ερωτήματα insert και select. Τι γίνεται όμως αν θέλουμε να τροποποιήσουμε μια ήδη υπάρχουσα εγγραφή; Γι' αυτή την εργασία προσφέρονται τα ερωτήματα update. Ένα τέτοιο ερώτημα ξεκινάει (προφανώς) με τον όρο update και συνεχίζει με το όνομα του πίνακα και τον όρο set. Μετά από αυτόν, ακολουθούν τα ονόματα των πεδίων που θέλουμε να τροποποιήσουμε, μαζί με τις αντίστοιχες τιμές. Προαιρετικά, μπορούμε να περιορίσουμε τις εγγραφές που θα τροποποιηθούν, ορίζοντας τα κατάλληλα κριτήρια. Αυτό γίνεται με τον όρο where και τον τρόπο που έχουμε ήδη γνωρίσει. Για παράδειγμα, έστω ότι θέλουμε να επέμβουμε στην ημερομηνία του μηνύματος με id=3 και να την αντικαταστήσουμε με την τρέχουσα ημερομηνία. Για να πετύχουμε κάτι τέτοιο αρκεί να εκτελέσουμε το εξής:

```
update messages set date=now() where id=3 ;
```

Αφαίρεση εγγραφών

Όπως διαπιστώνουμε καθημερινά, η καταστροφή είναι ευκολότερη από τη δημιουργία :) Για να αφαιρέσουμε εγγραφές από έναν πίνακα, αρκεί να χρησιμοποιήσουμε ένα ερώτημα delete. Σε αυτά τα ερωτήματα δηλώνουμε τον πίνακα στον οποίο θέλουμε να εργαστούμε και στη συνέχεια τα κριτήρια σύμφωνα με τα οποία θα διαγραφούν γραμμές. Έστω λοιπόν ότι επιθυμούμε να διαγράψουμε την εγγραφή με id = 3. Νομίζουμε ότι μπορείτε να φανταστείτε τη σύνταξη του σχετικού ερωτήματος:

```
delete from messages where id=3 ;
```

Η χρήση της delete —και ιδιαίτερα η χρήση του όρου where— απαιτεί τεράστια προσοχή. Ένα μικρό σφάλμα στη χρήση του όρου where μπορεί να προκαλέσει μεγάλη καταστροφή. Βλέπετε, αν για κάποιο λόγο αγνοηθεί ο όρος where και η MySQL εκτελέσει το υπόλοιπο ερώτημα, θα διαγραφεί ολόκληρος ο πίνακας!

Ψευδώνυμα πινάκων

Μια καλή τακτική ώστε να ξέρουμε ποια δεδομένα βρίσκονται πού, είναι να χρησιμοποιούμε περιγραφικά ονόματα για τους πίνακες. Αυτή η τακτική όμως οδηγεί σε μακροσκελή ονόματα, τα οποία δυσκολεύουν τη συγγραφή ερωτημάτων. Τη λύση σ' αυτό το ζήτημα αποτελούν τα ψευδώνυμα. Δείτε ένα παράδειγμα χρήσης και θα κατανοήσετε αμέσως την αξία τους:

```
select x.name, x.msg, x.date from messages as x where x.id > 1 ;
```

Όπως βλέπετε, μετά τον όρο from ακολουθεί ο πίνακας στον οποίο θέλουμε να εργαστούμε και στη συνέχεια εμφανίζεται ο όρος as, ακολουθούμενος από ένα ψευδώνυμο (το "x"). Με αυτόν τον τρόπο, οπουδήποτε θέλουμε να αναφερθούμε στον πίνακα messages, αρκεί να πληκτρολογήσουμε ένα "x". Η χρήση των ψευδωνύμων είναι χρήσιμη και στην περίπτωση που θέλουμε να ενώσουμε έναν πίνακα με τον εαυτό του. Μήπως σας φαίνεται περίεργο αυτό; Αναρωτιέστε σε τι θα ωφελούσε το να ενώσουμε έναν πίνακα με τον εαυτό του; Ε, λοιπόν, σας πληροφορούμε ότι αυτή η ενέργεια χρησιμοποιείται πολύ συχνά στις δικτυακές εφαρμογές. Ας δούμε ένα παράδειγμα:

Στο διαδίκτυο κυκλοφορεί ένα μεγάλο πλήθος από SPAM bots, τα οποία εντοπίζουν σελίδες με φόρμες εισαγωγής δεδομένων και τις χρησιμοποιούν για να αναρτήσουν διαφημιστικά μηνύματα. Αυτά τα bots, στην προσπάθειά τους να εμφανιστούν σαν φυσικά πρόσωπα, χρησιμοποιούν διάφορους έξυπνους αλγόριθμους ώστε να δημιουργούν αληθοφανή ονόματα χρηστών. Έτσι, σε πολλά φόρουμ βλέπουμε αναρτημένο το ίδιο μήνυμα SPAM, από αρκετούς διαφορετικούς χρήστες. Ας υποθέσουμε ότι έχουμε πρόσβαση στη βάση ενός φόρουμ το οποίο έχει γεμίσει με μηνύματα SPAM. Θέλουμε να τα εντοπίσουμε και για να τα καταφέρουμε θα αναζητήσουμε εγγραφές που έχουν διαφορετικό όνομα, αλλά το ίδιο μήνυμα. Κάτι τέτοιο μπορεί να γίνει με το ακόλουθο ερώτημα:

```
select x.name
from messages as x, messages as y
where x.msg = y.msg and x.name != y.name ;
```

Τροποποίηση πινάκων

Ουσιαστικά, το περιεχόμενο αυτής της ενότητας ανήκει στο τμήμα της γλώσσας που ονομάζεται DDL και με το οποίο ασχοληθήκαμε στο προηγούμενο άρθρο. Ωστόσο, η παρουσία του έχει περισσότερο νόημα εδώ. Πριν ξεκινήσουμε , πρέπει να τονίσουμε ότι η τροποποίηση πινάκων που περιέχουν ήδη δεδομένα απαιτεί μεγάλη προσοχή. Είναι πολύ εύκολο να γίνουν λάθη και να χαθούν πολύτιμα δεδομένα! Γι' αυτό το λόγο, κάθε φορά που θέλουμε να δημιουργήσουμε μια βάση δεδομένων, οφείλουμε να αφιερώνουμε αρκετό χρόνο στην προσεκτική "σχεδίαση" των πινάκων και των πεδίων. Πριν αρχίσουμε να δημιουργούμε οτιδήποτε, πρέπει να έχουμε ξεκαθαρίσει ποια δεδομένα θα διατηρούμε και πώς θέλουμε να είναι οργανωμένα.

Προκειμένου να τροποποιήσουμε έναν πίνακα, πρέπει να δημιουργήσουμε ένα ερώτημα που να ξεκινά με τη φράση "alter table". Αμέσως μετά πρέπει να δώσουμε το όνομα του πίνακα και στη συνέχεια να περιγράψουμε την επιθυμητή τροποποίηση. Ας δούμε τι είδους επεμβάσεις μπορούμε να κάνουμε στη δομή ενός πίνακα:

- Στην πιο απλή περίπτωση θα θέλουμε να προσθέσουμε μια επιπλέον στήλη. Κάτι τέτοιο μπορεί να γίνει με τη χρήση του όρου add, ο οποίος συντάσσεται ως εξής:

  ```
  add στοιχεία_νέας_στήλης [first | after όνομα_υπάρχουσας_στήλης]
  ```

  Αν δεν καθορίσουμε τη θέση της νέας στήλης (πριν ή μετά από κάποια άλλη) θα τοποθετηθεί στο τέλος του πίνακα. Με τον όρο "στοιχεία_νέας_στήλης" αναφερόμαστε σε μία ολοκληρωμένη *περιγραφή* των στοιχείων της νέας

στήλης (όνομα στήλης, τύπος δεδομένων και άλλες ειδικές παράμετροι[5]).

- Πολλές φορές, εξαιτίας του πρόχειρου αρχικού σχεδιασμού ή της αλλαγής της φύσης των δεδομένων, προκύπτει η ανάγκη να τροποποιήσουμε μια ήδη υπάρχουσα στήλη. Αυτό μπορεί να γίνει με δύο τρόπους: Αφενός, μπορούμε να χρησιμοποιήσουμε τον όρο change. Αμέσως μετά εισάγουμε το τρέχον όνομα της στήλης και στη συνέχεια όλα τα νέα στοιχεία της (όνομα, τύπος δεδομένων κ.ά). Αφετέρου, μπορούμε να χρησιμοποιήσουμε τον όρο modify. Μετά από αυτόν δίνουμε το όνομα το στήλης και στη συνέχει τα ανανεωμένα στοιχεία της. Αν και οι δύο τρόποι μοιάζουν πανομοιότυποι, σημειώστε ότι με τον όρο modify δεν μπορούμε να αλλάξουμε το όνομα της στήλης.
- Κάποιες άλλες φορές, θέλουμε να αφαιρέσουμε πλήρως μια στήλη. Αυτό μπορεί να γίνει με τον όρο drop, αμέσως μετά τον οποίο πρέπει να δώσουμε το όνομα της προς διαγραφή στήλης.

Ώρα για δουλειά...

Μπορεί να δημιουργήσαμε τη μικρή μας βάση δεδομένων με αγάπη, αλλά σε καμία περίπτωση δεν θα μπορούσαμε να τη χαρακτηρίσουμε ως υπόδειγμα καλής σχεδίασης. Κάθε άλλο, μάλιστα! Περιέχει μόνον έναν πίνακα, στη στήλη user του οποίου αποθηκεύεται το όνομα χρήστη που άφησε το εκάστοτε μήνυμα. Αυτό σημαίνει ότι αν ένας χρήστης αφήσει παραπάνω από ένα μήνυμα, το όνομά του θα εμφανίζεται πολλές φορές στη σχετική στήλη. Το γεγονός αυτό δυσκολεύει τον έλεγχο και τη διαχείριση των χρηστών που δημιουργούν μηνύματα. Για να απλοποιήσουμε τις εργασίες αυτού του είδους, θα ήταν προτιμότερο να δημιουργήσουμε έναν ξεχωριστό πίνακα, στον οποίο θα αποθηκεύονται μόνον οι χρήστες και τα στοιχεία τους, χωρίς τα μηνύματα. Κάτι τέτοιο θα μας επέτρεπε στο μέλλον να αναπτύξουμε περαιτέρω την εφαρμογή μας, προσθέτοντας νέες λειτουργίες και δυνατότητες. Και κάπως έτσι, φτάνουμε στο εξής δίλημμα: Να φτιάξουμε τη βάση και τους απαιτούμενους πίνακες από την αρχή ή να τροποποιήσουμε τον πίνακα που έχουμε ήδη; Η αλήθεια είναι ότι εφόσον το περιεχόμενο του πίνακα είναι δοκιμαστικό, θα ήταν καλύτερα να τον διαγράψουμε και να φτιάξουμε τα πάντα από την αρχή. Ωστόσο, κάνοντας κάτι τέτοιο, θα χάσουμε την ευκαιρία να δούμε στην πράξη όσα περιγράψαμε νωρίτερα. Για λόγους εξάσκησης, λοιπόν, θα επιλέξουμε τη λύση της τροποποίησης. Σηκώνουμε τα μανίκια, αλλά πριν αρχίσουμε τη σύνταξη ερωτημάτων SQL ξεκαθαρίζουμε το τι ακριβώς θέλουμε να πετύχουμε.

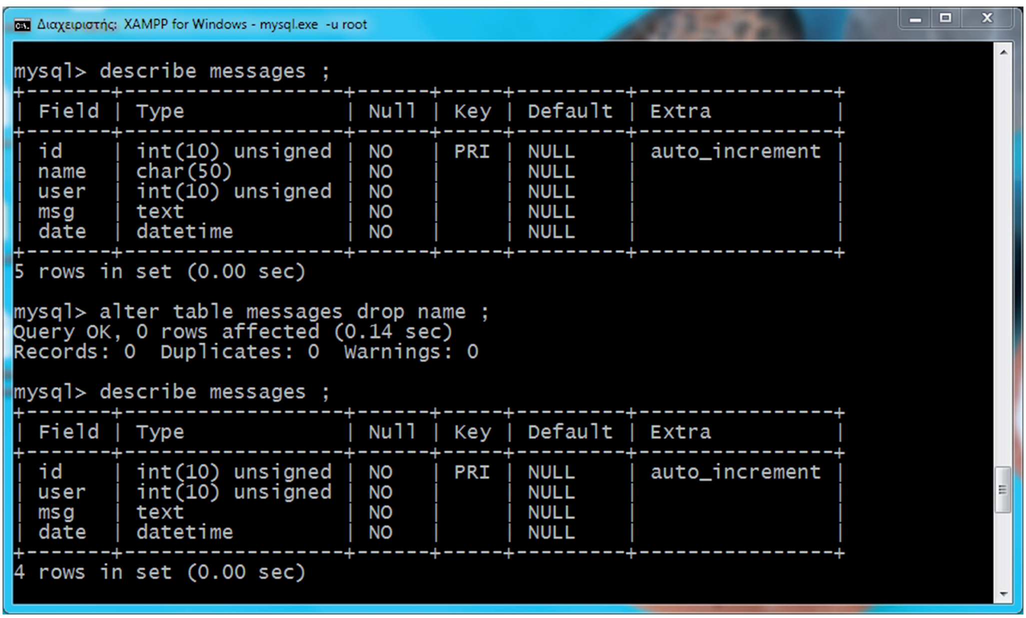
Αυτό που σκοπεύουμε να κάνουμε είναι να δημιουργήσουμε έναν νέο πίνακα, με το όνομα users. Σε αυτόν θα αποθηκεύονται τα στοιχεία των χρηστών: Το μοναδικό αναγνωριστικό (id) καθώς και ένα ψευδώνυμο (name). Στον πίνακα messages θα διαγράψουμε τη στήλη user και θα προσθέσουμε μία νέα, με το όνομα userid. Σε αυτήν θα καταχωρείται το id του χρήστη που άφησε το εκάστοτε μήνυμα. Παρακάτω φαίνονται τα ερωτήματα που πετυχαίνουν όλα όσα περιγράψαμε, ενώ ταυτόχρονα διατηρούν ανέπαφα και τα υπάρχοντα δεδομένα:

```
create table visitors(
    id int unsigned not null auto_increment primary key,
    name char(50) not null
) ;
```

5. Αν θέλετε να φρεσκάρετε τη μνήμη σας, ανατρέξτε στις ενότητες "δημιουργία πινάκων", "τύποι πεδίων" και "ιδιότητες πεδίων", στο προηγούμενο άρθρο της σειράς.

```
insert into visitors (name) select distinct name from messages ;
alter table messages add visitor int unsigned not null after name ;
update messages, visitors
set messages.userid = visitors.id
where messages.name = visitors.name ;
alter table messages drop name ;
```

Ξεκινάμε με τη δημιουργία του πίνακα visitors, ο οποίος θα διατηρεί τα στοιχεία των χρηστών. Η στήλη με τα ονόματα (το πεδίο του ονόματος, δηλαδή) φροντίζουμε να έχει το ίδιο μήκος με την αντίστοιχη στήλη του πίνακα messages. Αυτό το κάνουμε ώστε να μεταφέρουμε τα δεδομένα χωρίς να υπάρξουν απώλειες.



Τα στοιχεία του πίνακά μας, πριν και μετά την εκτέλεση του ερωτήματος τροποποίησης.

Αμέσως μετά, εκτελούμε ένα ερώτημα εισαγωγής (insert), για να προσθέσουμε στον πίνακα visitors τα ονόματα των χρηστών που βρίσκονται αποθηκευμένα στον πίνακα messages. Αυτό το ερώτημα, όμως, είναι αρκετά διαφορετικό απ' όσα έχουμε δει μέχρι στιγμής. Τα δεδομένα που πρόκειται να εισαχθούν, προέρχονται από ένα δεύτερο ερώτημα (select). Με το δεύτερο ερώτημα, λαμβάνουμε τα ονόματα από τον πίνακα messages. Με τη χρήση του όρου distinct, εξασφαλίζουμε ότι δεν θα μεταφερθούν τα διπλότυπα ;)

Στη συνέχεια προσθέτουμε μια νέα στήλη στον πίνακα messages, με το όνομα userid. Οι ιδιότητες αυτής της στήλης φροντίζουμε να είναι συμβατές με εκείνες της στήλης id, του πίνακα visitors. Αυτό το κάνουμε διότι οι τιμές του πεδίου userid θα προέρχονται ακριβώς από το πεδίο id του πίνακα visitors. Όπως καταλαβαίνετε, μέσω αυτής της στήλης θα πραγματοποιούμε μια συσχέτιση μεταξύ των δύο πινάκων. Προσέξτε ότι ο όρος after, τοποθετεί τη νέα στήλη μετά από τη στήλη name.

Αυτό που μένει να κάνουμε τώρα είναι να ενημερώσουμε όλες τις εγγραφές του πίνακα messages και συγκεκριμένα να αντιστοιχίσουμε σ' αυτές τα σωστά id. Αυτό επιτυγχάνεται με το ερώτημα update, στο οποίο συμμετέχουν πεδία κι από τους δύο πίνακες! Τέλος, αφού δημιουργήσαμε όλες τις νέες δομές και μεταφέραμε τα δεδομένα, μπορούμε να διαγράψουμε τη στήλη name από τον πίνακα messages. Αυτό επιτυγχάνεται με το ερώτημα drop.

Όπως βλέπετε, η τροποποίηση των πινάκων είναι μια σχετικά περίπλοκη εργασία, αλλά δεν είναι αδύνατη. Προκειμένου πάντως να αποφεύγουμε τις άσκοπες περιπλανήσεις στα τρικυμιώδη νερά της MySQL, είναι καλό να αφιερώνουμε λίγο περισσότερο χρόνο κατά την αρχική σχεδίαση των πινάκων και των πεδίων.

Αντίγραφα ασφαλείας

Η τακτική λήψη backup αποτελεί μια εργασία μείζονος σημασίας, την οποία δεν πρέπει να αμελούμε ποτέ. Ο νόμος του Μέρφυ καιροφυλακτεί και μπορεί να δημιουργήσει προβλήματα ανά πάσα στιγμή! Ευτυχώς, η MySQL διευκολύνει αρκετά στη λήψη αντιγράφων ασφαλείας — και μάλιστα παρέχει αρκετά σχετικά εργαλεία. Η μέθοδος που προτιμά ο γράφων περιλαμβάνει τη χρήση του εργαλείου mysqldump. Με τη βοήθειά του μπορούμε να πάρουμε ένα πλήρες αντίγραφο μιας βάσης δεδομένων ή και όλων των βάσεων που ενδέχεται να έχουμε δημιουργήσει. Το αποτέλεσμα από την εκτέλεση του εν λόγω εργαλείου εμφανίζεται στη γραμμή εντολών (στην τυπική έξοδο) και ως εκ τούτου, μπορούμε να το ανακατευθύνουμε σε ένα αρχείο. Παρακάτω φαίνονται δυο απλά παραδείγματα χρήσης του mysqldump:

```
mysqldump --u=onoma_christi  onoma_basis > backup_file_name.sql
mysqldump --u=onoma_christi  --all-databases > backup_file_name.sql
```

Εναλλακτική διαχείριση

Μετά από όσα έχουμε δει και με την απαραίτητη εξάσκηση που σας παροτρύνουμε να κάνετε, θα μπορείτε να διαχειριστείτε μια βάση δεδομένων και τους πίνακές της από τη γραμμή εντολών. Αφού κατακτήσετε αυτή την κρίσιμη δεξιότητα, μπορείτε να στραφείτε και σε μια εναλλακτική λύση για τη διαχείριση των δεδομένων σας, η οποία δεν απαιτεί πληκτρολόγηση. Αναφερόμαστε στο εργαλείο phpMyAdmin, το οποίο μέσα από τον web browser παρέχει ένα εύχρηστο περιβάλλον για τη διαχείριση βάσεων. Πάντως, πριν καταφύγετε σε κάποιο εργαλείο αυτού του είδους, πρέπει να σας πούμε ότι η χρήση τους απαιτεί μεγάλη προσοχή. Το ίδιο ισχύει και για την εγκατάσταση των υπό συζήτηση εργαλείων, όπως επίσης και για τη ρύθμισή τους. Βλέπετε, ένα τέτοιο εργαλείο που δεν έχει ρυθμιστεί σωστά ή που δεν είναι ενημερωμένο, ενδέχεται να δημιουργήσει σημαντικά κενά στην ασφάλεια της βάσης κι ενδεχομένως στην ασφάλεια ολόκληρου του server!

Μέχρι το επόμενο τεύχος σας προτείνουμε να πειραματιστείτε όσο πιο πολύ μπορείτε, με όλα όσα έχουμε μάθει σχετικά με τη MySQL. Επιπρόσθετα, αν το τραβάει η όρεξή σας, μπορείτε να εμβαθύνετε ακόμα περισσότερο στη χρήση της, μελετώντας την τεκμηρίωση στη δικτυακή της κατοικία: www.mysql.org. Ετοιμαστείτε για το επόμενο άρθρο, στο οποίο θα μελετήσουμε τη διασύνδεση της MySQL με τα υπόλοιπα υποσυστήματα που απαρτίζουν το LAMP stack. Μετά κι από αυτό, θα είμαστε σε θέση να δημιουργήσουμε σύγχρονες δικτυακές εφαρμογές, με δυναμικό περιεχόμενο!

Skill: Intermediate
Tags: pfSense, bridging, gateways, multi-WAN, load balancing, failover, PF, NAT



Ζηλευτός υπέρ-router, με υποστήριξη multi-WAN!

Κάποιες φορές η επικαιρότητα είναι σαν να γράφει η ίδια τους προλόγους των άρθρων μας — και φυσικά δεν κάνει μόνο σ' εμάς τη χάρη. Μερικές βδομάδες πριν, αφού είχαμε ήδη τελειώσει με τη δημιουργία ενός πανίσχυρου router κυρίως από την ανακύκλωση παλιού εξοπλισμού, αποφασίσαμε ότι οπωσδήποτε θα έπρεπε να σας πούμε για το εγχείρημα. Καθίσαμε λοιπόν μια μέρα και κάναμε τη διάρθρωση του άρθρου που τώρα διαβάζετε. Πολύ σύντομα, μπροστά μας βρήκαμε το πάντα τρισμέγιστο πρόβλημα της εισαγωγής: Τι να πρωτογράψουμε, ώστε αφενός να σας κεντρίσουμε το ενδιαφέρον κι αφετέρου να δείτε αμέσως ότι αξίζει κι εσείς ν' ασχοληθείτε; Κι εκεί που προβληματιζόμασταν κι αυτο-μπλοκαριζόμασταν, ήλθε η επικαιρότητα να μας δώσει ένα φιλικό χτύπημα στον ώμο...

Ο πρόσφατος εντοπισμός του αυτο-αναπαραγόμενου worm ονόματι TheMoon, το οποίο αναζητά και μολύνει μοντέλα router της Linksys, δεν είναι το μοναδικό περιστατικό που προκάλεσε μια κάποια ανησυχία σε όσους ενδιαφέρονται στοιχειωδώς περί θεμάτων ασφαλείας. Παρόμοια προβλήματα έχουν κατά καιρούς ακουστεί και για routers των Asus, D-Link και Netgear, για ν' αναφέρουμε μερικά μόνο από τα γνωστά ονόματα. Κάτοχοι routers της Asus, για παράδειγμα, βρήκαν πριν μερικές βδομάδες μυστηριώδη text files σε USB δίσκους που ήταν συνδεδεμένοι στις συσκευές. Τα αρχεία δεν τα είχαν δημιουργήσει οι ίδιοι. Στο περιεχόμενό τους υπήρχαν αναφορές σε αδυναμίες, οι οποίες επιτρέπουν σε οποιονδήποτε χρήστη του Internet ν' αποκτά πρόσβαση στα δίκτυα πίσω από τους ευπαθείς routers.

Γενικά, τα θέματα ασφαλείας στους δρομολογητές για το σπίτι ή το γραφείο (SOHO routers), μόνο κάτι το πρωτάκουστο δεν είναι. Αν επιχειρούσαμε να κάνουμε μια πρόχειρη κατηγοριοποίηση των αδυναμιών, τότε θα καταλήγαμε με μια λίστα που σίγουρα θα περιελάμβανε τα ακόλουθα:

- bugs στο firmware που ευθύνονται για ανοικτά ports, μέσω των οποίων είναι δυνατή η διαχείριση —ακόμη κι από το WAN interface— της συσκευής, χωρίς την εισαγωγή credentials
- backdoors στο firmware, τα οποία στην καλύτερη περίπτωση είναι ξεχασμένα εκεί από τον κατασκευαστή, ενώ στη χειρότερη είναι επίτηδες "φυτεμένα"
- ευπαθείς υπηρεσίες, π.χ., διαμοίρασης αρχείων κι εκτυπωτών
- προβληματικές υλοποιήσεις πρωτοκόλλων όπως, π.χ., UPnP
- επιπόλαια configurations, για χάριν της ευκολίας χρήσης
- firewalls που αποτυγχάνουν να κάνουν τη δουλειά τους

Θα μπορούσαμε να συνεχίσουμε, αλλά ήδη βλέπετε ξεκάθαρα την εικόνα που έχει σχηματιστεί. Σκεφτείτε τώρα το modem/router που κάθεται ήσυχο ήσυχο στο σπίτι, ανοικτό 24 ώρες το 24ωρο και σας εξυπηρετεί σιωπηρά κι αδιαμαρτύρητα. Αλήθεια, πότε ήταν η τελευταία φορά που το αναβαθμίσατε; Ο κατασκευαστής, άραγε, συνεχίζει να το υποστηρίζει, κυκλοφορώντας patches με bug fixes και security updates; Ακόμη κι αν το κάνει κι εσείς δεν αμελείτε να φρεσκάρετε το firmware της συσκευής, πόση πίστη έχετε στην καλή κι επιμελή εργασία που κάνουν οι προγραμματιστές της εταιρείας; Εδώ που τα λέμε, κατά πάσα πιθανότητα οι άνθρωποι είναι ικανότατοι και κάνουν μια χαρά τη δουλειά τους. Έχουν ωστόσο κι αυτοί δικαίωμα στο λάθος, οπότε στην πραγματικότητα άλλο είναι το ερώτημα: Από εκείνους που ανακαλύπτουν προβλήματα ασφαλείας στα firmware των modem/routers, πόσοι τα αναφέρουν στην κατασκευάστρια εταιρεία και δεν τα εκμεταλλεύονται προς ίδιον όφελος;

Εντάξει, δεν υπάρχει λόγος ν' αγχώνεστε. Εμείς απλά μερικές σκέψεις κάνουμε — και τις μοιραζόμαστε μαζί σας.

Όχι άλλη ανησυχία

Αντί να κάνουμε απλά "μερικές σκέψεις" και ν' ανησυχούμε γενικώς, μπορούμε να πάρουμε την κατάσταση στα χέρια μας. Διπλή είναι η δική μας προσέγγιση στο θέμα.

- Θεωρούμε δεδομένο ότι το modem/router μας είναι ευπαθές — και είναι ευπαθές ακόμη κι αν η κατασκευάστρια εταιρεία κυκλοφορεί τακτικά firmware updates κι εμείς τα εγκαθιστούμε το αργότερο μισή ώρα μετά τη σχετική ανακοίνωση. Ακριβώς γι' αυτό, φροντίζουμε κατ' αρχάς ώστε να έχουμε απενεργοποιημένα τα όποια extra features του router (και το WPS είναι το πρώτο που μας έρχεται τώρα στο μυαλό: http://deltahacker.gr/?p=7029). Απενεργοποιημένο είναι επίσης το UPnP. Φυσικά, φροντίζουμε ώστε υπηρεσίες όπως FTP, SMB κ.λπ. να είναι επίσης ανενεργές. Δεν ξέρουμε για εσάς, εμείς όμως ποτέ δεν θα παίρναμε ένα router με θύρες USB, ώστε να μοιράσουμε δίσκους κι εκτυπωτές στο τοπικό δίκτυο. Μόνη περίπτωση να το κάναμε θα 'ταν αν είχαμε συσκευή συμβατή με OpenWrt ή/και DD-WRT[1], όπου τότε πρώτα θα εγκαθιστούσαμε το αντίστοιχο Open Source firmware και μετά θα ενεργοποιούσαμε παραφερνάλια όπως, π.χ., file και printer sharing services.
- Όποτε είναι δυνατόν, έχουμε ένα δικό μας box σε ρόλο router/firewall. Σε ένα τέτοιο σενάριο, οι clients είναι πίσω από το box και μέσω αυτού βγαίνουν στον έξω κόσμο. Έχουμε, δηλαδή, το τοπικό δίκτυο με τους clients που μιλάνε (ενσύρματα ή ασύρματα) στον δικό μας DIY[2] router, εκείνος με τη σειρά του μιλάει με το modem που μας έδωσε ο ISP και το προαναφερθέν απλά φροντίζει να διατηρεί το link με τον ISP.

Το καλό με τον DIY router είναι ότι μπορούμε να επιλέξουμε το λειτουργικό του σύστημα το οποίο, κατά κανόνα, θα 'ναι περί τις 129 φορές ασφαλέστερο από το OS που τρέχει το όποιο modem/router. (Επιπρόσθετα, το λειτουργικό του δικού μας router θα μας επιτρέπει να κάνουμε πράγματα που με το απλό modem/router δεν θα δικαιούμαστε καν να ονειρευτούμε.) Το κακό με τη συγκεκριμένη προσέγγιση είναι ότι, αρχικά τουλάχιστον, οφείλουμε να επενδύσουμε λίγο χρόνο —όχι κατ' ανάγκη και χρήμα— για το στήσιμο του box, ενώ εξυπακούεται κι ότι οφείλουμε να ξέρουμε τι κάνουμε. Άρα, υπό αυτή την έννοια, η ευθύνη για την προστασία του τοπικού δικτύου βαραίνει αποκλειστικά εμάς. Εδώ που τα λέμε, όμως, από τη στιγμή που διαβάζετε το deltaHacker θεωρούμε δεδομένο ότι *δεν* διακατέχεστε στο ελάχιστο από παρόμοιες ευθυνοφοβίες. Αντίθετα, στοιχηματίζουμε ότι αυτή τη στιγμή βλέπετε κι εσείς άλλη μια ευκαιρία για πειραματισμό, μάθηση και φυσικά *αληθινή* διασκέδαση.

Το setup που έχουμε και προτείνουμε

Όσοι μας διαβάζετε τακτικά, πιθανώς θα ξέρετε ήδη ότι για το λειτουργικό του router μας είχαμε να επιλέξουμε μεταξύ OpenBSD και pfSense. Τελικά μάς κέρδισε το όμορφο και χρηστικό web panel που προσφέρει το pfSense, οπότε πήγαμε μ' αυτό.

Στις βασικές υπηρεσίες που παρέχει ο router μας συγκαταλέγονται ο caching nameserver κι ο DHCP server, ενώ φυσικά υπάρχει και το PF firewall. Μια άλλη, εξαιρετικά χρήσιμη υπηρεσία είναι αυτή που προσφέρεται από το OpenVPN. Έχοντας ρυθμίσει το pfSense box ώστε να λειτουργεί ως OpenVPN server, τότε όπου κι αν βρισκόμαστε μπορούμε να συνδεόμαστε, μέσω ενός ασφαλούς τούνελ, στο τοπικό

1. Βλέπε σχετικά άρθρα στα τεύχη 011, 012 και 015.
2. Do It Yourself.

δίκτυο του σπιτιού. Ποσώς μας ενδιαφέρει αν το σημείο (της αρχικής) σύνδεσης είναι επισφαλές, όπως, π.χ., συχνά συμβαίνει με τα ανοικτά, ελεύθερα HotSpots. Πράγματι, το κανάλι επικοινωνίας από τον υπολογιστή ή τη συσκευή μας μέχρι και το pfSense box είναι ισχυρά κρυπτογραφημένο και η δικτυακή κίνηση περνάει μέσα από αυτό. Προσέξτε: Από το εν λόγω κανάλι περνάει όλη η δικτυακή κίνηση ανεξαιρέτως κι όχι μόνον εκείνη που αφορά στο LAN, το οποίο προστατεύει το pfSense box. Πρακτικά, δηλαδή, στο Internet δεν βγαίνουμε από τον ξένο router που βρίσκεται, π.χ., πίσω από το δημόσιο HotSpot, αλλά από το pfSense box που έχουμε στο σπίτι. Θα το πούμε και διαφορετικά. Μπορεί να είμαστε στο Βλαδιβοστόκ, άπαξ όμως και συνδεθούμε στον OpenVPN server του pfSense τότε για το Internet ολάκερο θα 'ναι σαν να βρισκόμαστε στο σπιτάκι μας (στην Καλαμαριά).

Τώρα, το δικό μας pfSense box έχει τέσσερις κάρτες Ethernet. Οι δύο εξ αυτών είναι γεφυρωμένες και χρησιμοποιούνται από τους πελάτες του LAN. Οι άλλες δύο μιλάνε με δύο διαφορετικά modem/router, το ένα από τον VDSL provider και το άλλο από τον cable provider που χρησιμοποιούμε. Ναι, σωστά καταλάβατε: Ο router μας έχει δύο συνδέσεις με το Internet κι αυτό εκτός από redundancy σημαίνει και κατά πολύ υψηλότερες ταχύτητες download/upload. Αλλά περισσότερα επ' αυτού σε πολύ λίγο, όπου θα δείξουμε πώς επιτυγχάνεται το λεγόμενο multi-WAN μέσω του pfSense.

Στα "συν" του pfSense συγκαταλέγονται και τα packages. Πρόκειται για εφαρμογές που είναι έτοιμες προς εγκατάσταση και χρήση, στο επίπεδο του router. Κλασικό παράδειγμα package που θα θέλετε να εγκαταστήσετε είναι αυτό που αφορά στο Squid (web proxy cache). Μια άλλη, εξαιρετικά ενδιαφέρουσα εφαρμογή που παρέχεται ως package, είναι το Snort. Πρόκειται για ένα πανίσχυρο σύστημα intrusion detection και prevention το οποίο, αν το box σας έχει αρκετή μνήμη, αξίζει να το εγκαταστήσετε και να το δείτε σε δράση.

Περισσότερα για τα OpenVPN, Squid και Snort θα συζητήσουμε σε επόμενα άρθρα μας — ενδεχομένως και σε κάποιο επεισόδιο του deltaCast (http://deltahacker.gr/category/deltaCast). Προς το παρόν, ελάτε να εξετάσουμε μια βασική εγκατάσταση του pfSense και φυσικά να δούμε πώς "σετάρουμε" το multi-WAN. Μετά απ' αυτό το τελευταίο, κάτι μας λέει ότι θα ελέγξετε τα πάγια μηνιαία έξοδά σας, κοιτάζοντας τι μπορείτε να κάνετε ώστε να έχετε κι εσείς δύο συνδέσεις ίντερνετ στο σπίτι. Αλλά μπορεί και να κάνουμε λάθος :P

Το hardware και η εγκατάσταση

Πρακτικά, η μόνη αδιαπραγμάτευτη προϋπόθεση από πλευράς hardware είναι υπολογιστής με δύο τουλάχιστον κάρτες δικτύου: μία στην οποία θα συνδέονται οι clients (LAN port)[3] καθώς κι άλλη μία, για να επικοινωνεί το pfSense με το modem/router (WAN port).

Εδώ γύρω, στα κεντρικά της Parabing Creations στο Northnern Outpost, μέχρι πρότινος υπήρχε ένα box σε ρόλο fileserver. Το λειτουργικό του ήταν το FreeBSD — και φυσικά είχε το ZFS του. Αν κι ευχαριστημένοι από τις υπηρεσίες που προσέφερε το μηχάνημα, από καιρό είχαμε αποφασίσει ότι πρέπει ν' αλλάξουμε τη στρατηγική των

3. Ενδεχομένως μέσω κάποιου switch ή/και wireless access point.

backups και να φτιάξουμε κάτι που άνετα θα μπορούμε να ονομάζουμε "personal cloud"[4]. Πράλληλα, είχαμε βαρεθεί να θαυμάζουμε και να παίζουμε με το pfSense *μόνο* σε VM, οπότε καταλαβαίνετε ότι πλέον δεν κρατιόμασταν και το θέλαμε *και* σε αληθινό υπολογιστή.

Έτσι, το πρώτο πράγμα που κάναμε ήταν να πάρουμε backup και μετά ν' απαλλάξουμε το PC-fileserver από τους τέσσερις μαγνητικούς δίσκους που διέθετε[5]. Τώρα, αν και το pfSense μπορεί κάλλιστα να εγκατασταθεί σε φλασάκι USB ή σε κάρτα Compact Flash, από τη στιγμή που οπωσδήποτε θα ασχολούμασταν με Squid, Snort κι άλλες εφαρμογές, η εγκατάσταση σε δίσκο ήταν μονόδρομος. Αγοράσαμε λοιπόν έναν SSD, χωρητικότητας 64GB. Μπορεί ν' ακούγεται μικρός, ωστόσο είναι υπεραρκετός για το pfSense, τις έξτρα εφαρμογές, την cache του Squid, τα log files κ.ο.κ. Εκτός από την υψηλή ταχύτητα, στα θετικά του SSD συγκαταλέγονται η χαμηλή κατανάλωση και η συμβολή του προς ένα αθόρυβο box.

Ο επεξεργαστής του υπολογιστή, εξάλλου, είναι ένας 64μπιτος Intel Core 2 Duo στα 1,6GHz (διπύρηνος) και, περιττό να τονίσουμε, για τις ανάγκες μας η ισχύς του περισσεύει. Αρκετή ήταν και η μνήμη του box, χωρητικότητας 2GB (τύπου DDR2-666), η οποία έφτανε ακόμη και για το Snort. Και ναι, η χρήση του παρελθοντικού χρόνου γίνεται ακριβώς για το λόγο που υποψιάζεστε: Δεν κρατηθήκαμε και, χωρίς να υπάρχει ιδιαίτερος λόγος, εφοδιάσαμε το σύστημα με άλλα 2GB RAM, φτάνοντας έτσι στο άνω όριο που υποστηρίζει η μητρική (4GB).

Πριν τη μετατροπή του σε router, το box επικοινωνούσε με τον έξω κόσμο μέσω ενός ενσωματωμένου στη μητρική ελεγκτή της RealTek (8168/8111 PCIe Gigabit Ethernet). Του προσθέσαμε μια Intel Gigabit Ethernet PRO/1000 τύπου PCI Express, καθώς κι άλλες δύο D-Link DGE-528(T) τύπου PCI (σκέτο), επίσης Gigabit Ethernet. Όλες αυτές τις κάρτες τις είχαμε από παλιότερα συστήματα, ξεχασμένες σ' ένα συρτάρι. Το γιατί από τις δύο κάρτες δικτύου που απαιτούνται πήγαμε στις τέσσερις, θα φανεί στη συνέχεια.

Τώρα, για το στάδιο της εγκατάστασης του pfSense δεν έχουμε να πούμε κάτι. Σημειώνουμε μόνο ότι συνδέσαμε προσωρινά στο box πληκτρολόγιο κι οθόνη, και μετά του περάσαμε την τελευταία έκδοση (2.1) της εκδοχής 64bit του λειτουργικού. Προηγουμένως, είχαμε κατεβάσει από το

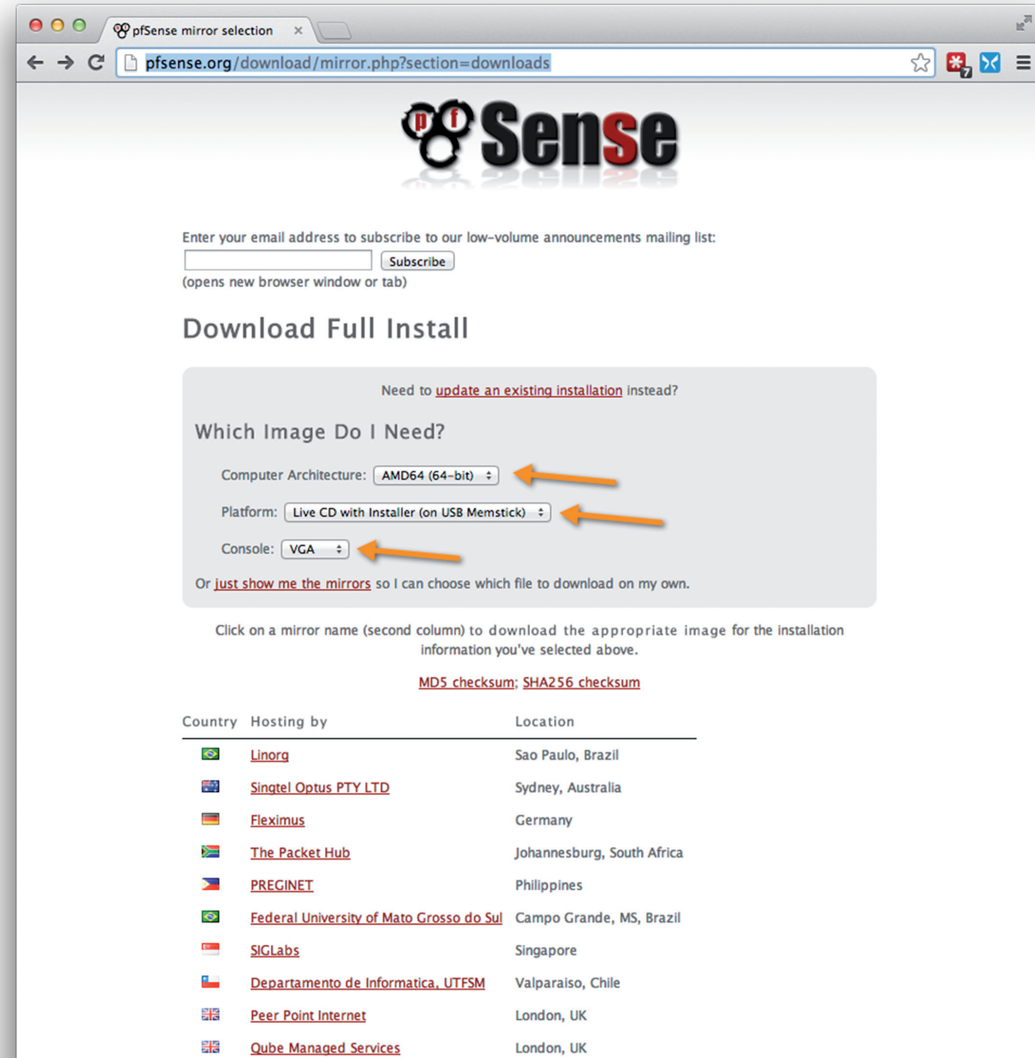
`http://pfsense.org/download/mirror.php?section=downloads`

το image του pfSense που είναι κατάλληλο για εγκατάσταση από USB stick. Όσοι πάντως ενδιαφέρονται για τις λεπτομέρειες της διαδικασίας ή/και δεν έχουν δει ξανά το pfSense σε δράση, παροτρύνονται να παρακολουθήσουν τώρα το 7ο επεισόδιο της 1ης σεζόν του deltaCast:

`http://deltahacker.gr/deltacast-s01e07`

Αν και αφορά σε παλαιότερη έκδοση του pfSense, λίγα έχουν αλλάξει σε σχέση με την τρέχουσα.

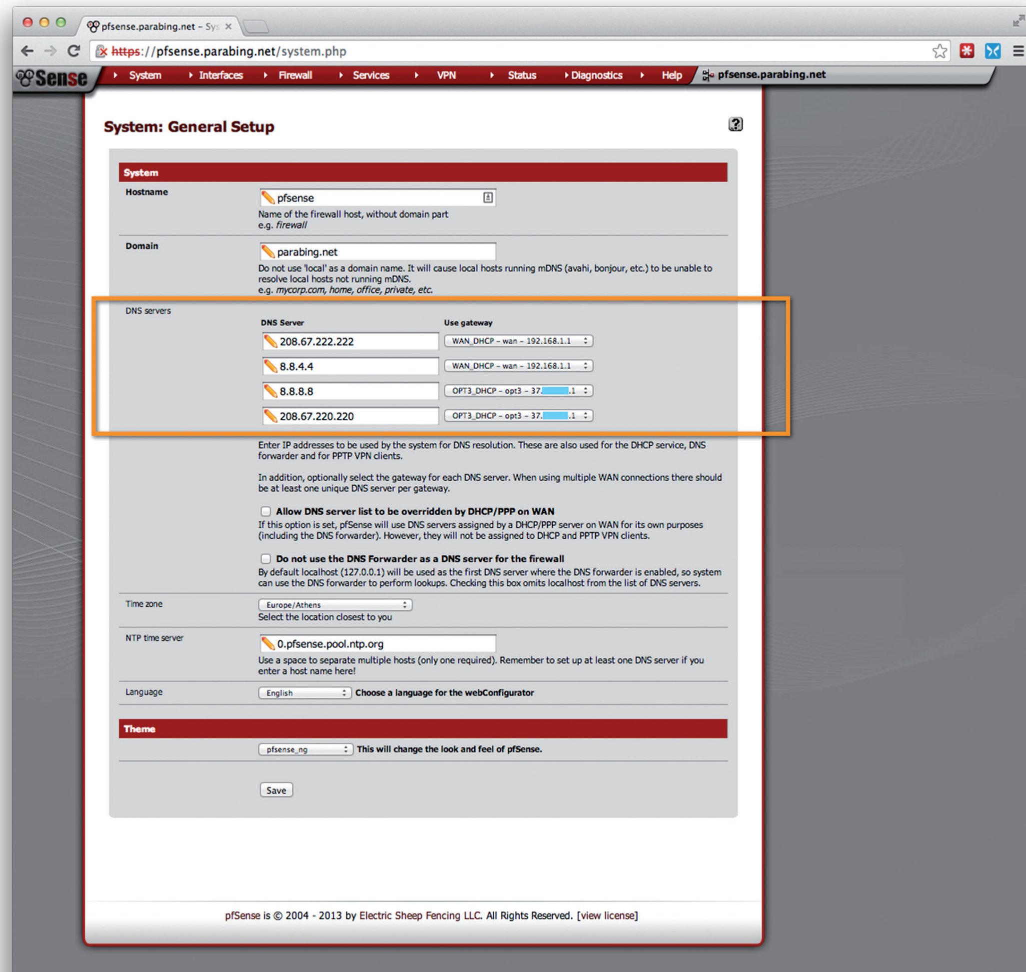
4. Περισσότερα γι' αυτό το πρότζεκτ θα πούμε σε επόμενα τεύχη.
5. Διαφορετικών χωρητικοτήτων, με τον μεγαλύτερο να έχει μέγεθος 1,5TB κι όλοι μαζί να μην ξεπερνούν τα 2,4TB.



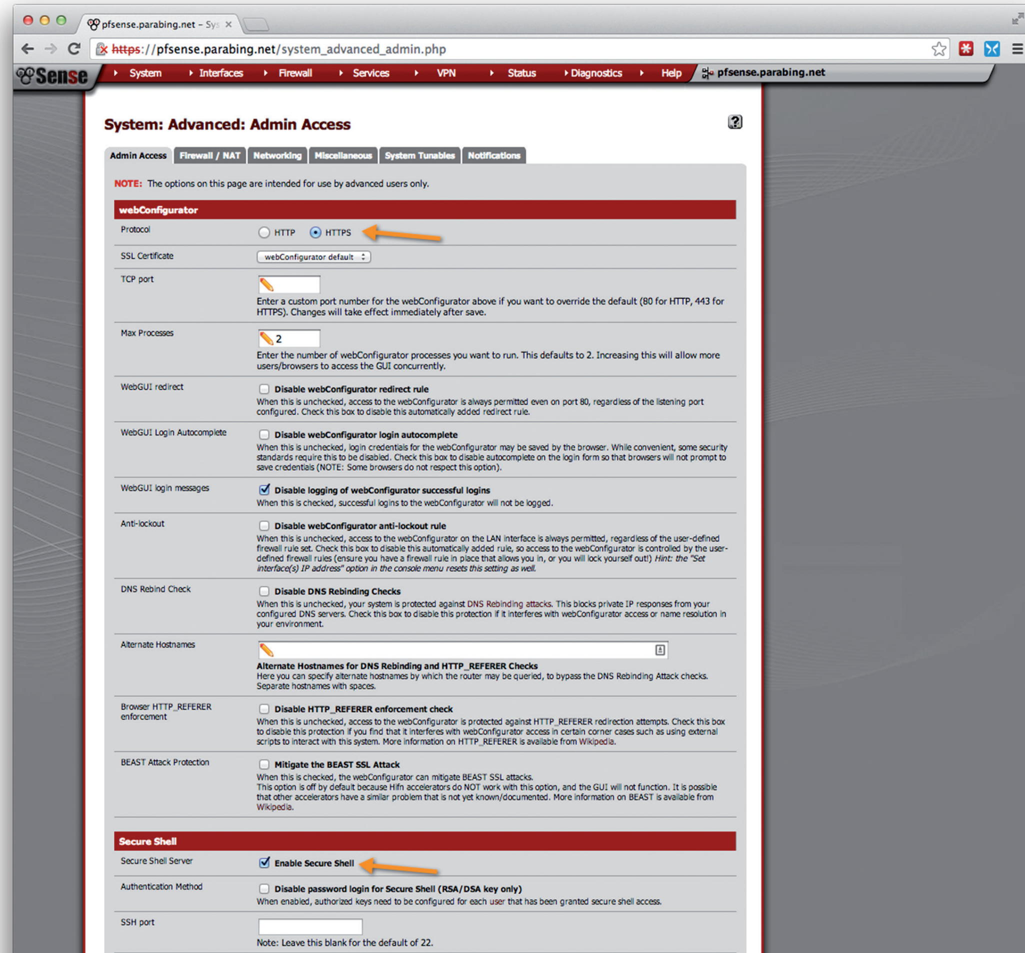
Το box μας φυσικά και *δεν* έχει οδηγό CD/DVD, οπότε για την εγκατάσταση του pfSense κατεβάσαμε το Live CD image που είναι κατάλληλο για εγγραφή σε USB stick (πλήρες όνομα αρχείου: pfSense-memstick-2.1-RELEASE-amd64.img.gz). Μετά το download και την αποσυμπίεση, η εγγραφή του image γίνεται με ένα εργαλείο όπως το dd (Linux / *BSD / OS X) ή το physdiskwrite (Windows, http://m0n0.ch/wall/physdiskwrite.php)

Όψεις του pfSense

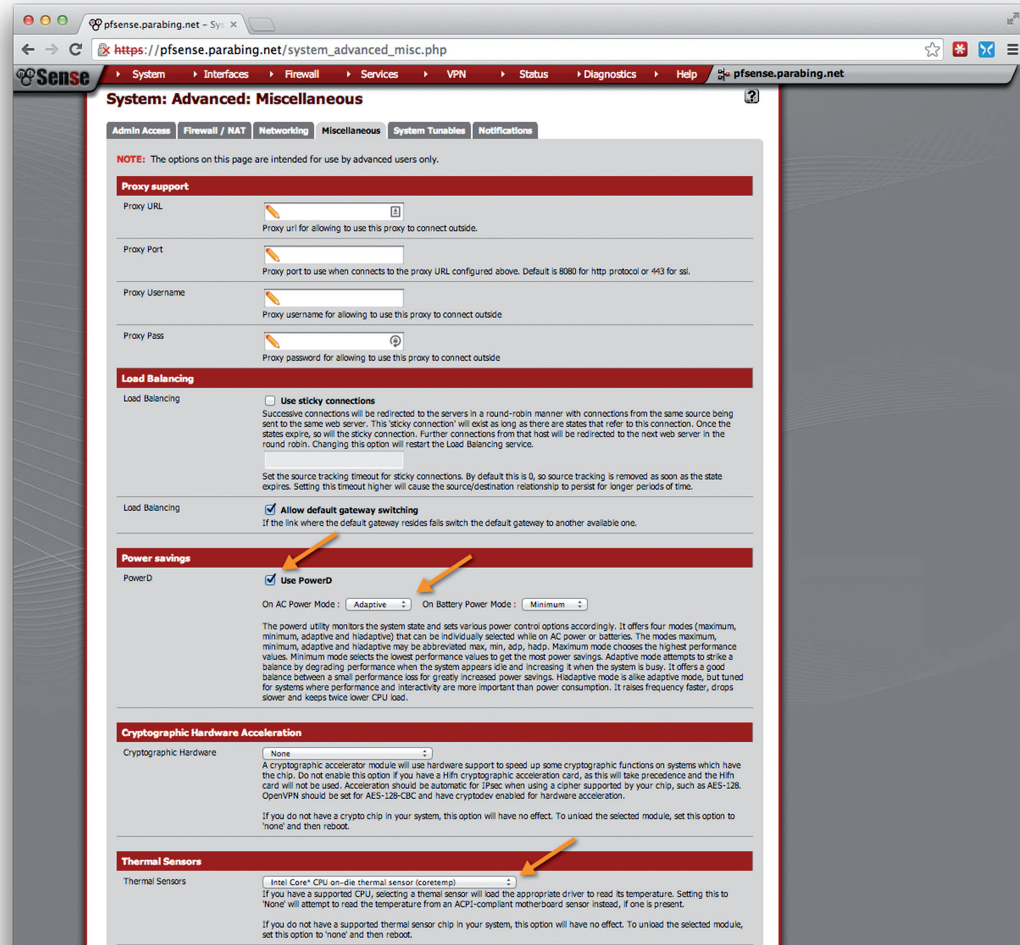
Αυτό που κατά κύριο λόγο θέλουμε να δείξουμε στο παρόν άρθρο είναι η ρύθμιση του multi-WAN. Προηγουμένως, όμως, είναι καλή ιδέα να σας κάνουμε μια σύντομη ξενάγηση στο web panel του δικού μας pfSense box, και να σας παρουσιάσουμε τη λογική πίσω από κάποιες βασικές επιλογές και ρυθμίσεις. Δείτε λοιπόν τις εικόνες από την 1 έως και τη 10 και διαβάστε τις αντίστοιχες περιγραφές.



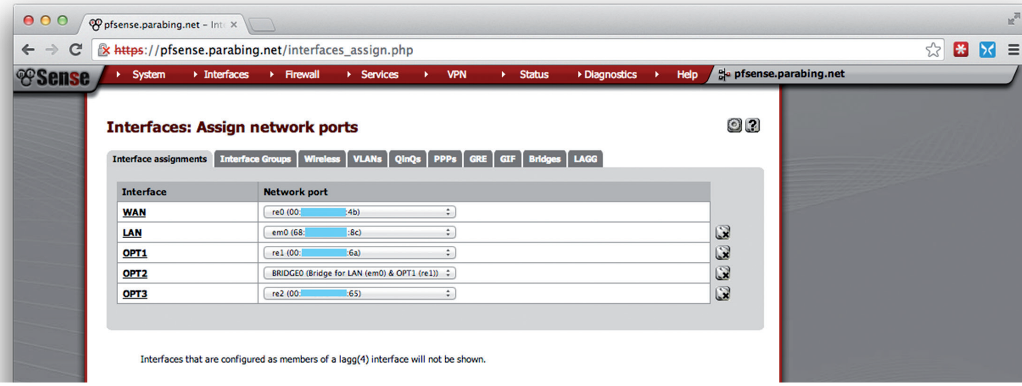
Εικόνα 1 (System → General Setup). Το box μας χρησιμοποιεί τους δημόσιους nameservers των Google (8.8.8.8, 8.8.4.4) και OpenDNS (208.67.222.222, 208.67.220.220). Σημειώστε ότι υπάρχουν δύο συνδέσεις WAN και συνεπώς δύο WAN gateways, όπου σε καθένα εξ αυτών έχουμε αντιστοιχίσει από έναν nameserver *κάθε* εταιρείας. Τους έχουμε μοιράσει, δηλαδή, στα δύο WAN gateways που έχει ο router μας.



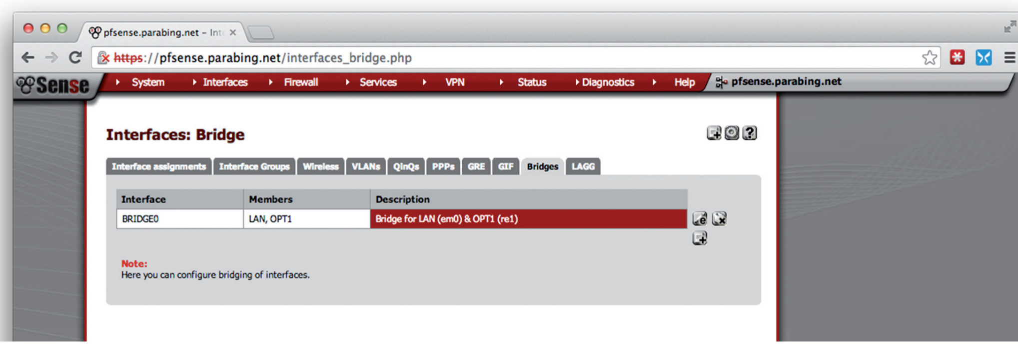
Εικόνα 2 (System → Advanced → Admin Access). Αν και το οικιακό LAN στο οποίο κατοικεί το pfSense είναι κατά 99,129% φιλικό, προτιμούμε να συνδεόμαστε στο web panel του router μέσω HTTPS κι όχι μέσω του απλού, μη-κρυπτογραφημένου HTTP. Επαναλαμβάνουμε ότι ο κίνδυνος για sniffing είναι πρακτικά ανύπαρκτος, ωστόσο Ποτέ Δεν Ξέρεις (TM). Παρατηρήστε εξάλλου ότι στο pfSense box έχουμε ενεργοποιημένη την υπηρεσία του SSH (για το τοπικό δίκτυο). Ίσως χρειαστεί να συνδεθούμε στην κονσόλα του router, π.χ., επειδή πρέπει να τον επανεκκινήσουμε και για κάποιον λόγο δεν μπορούμε από το web panel. Απίθανο σενάριο, ωστόσο και πάλι ισχύει το Ποτέ Δεν Ξέρεις (TM).



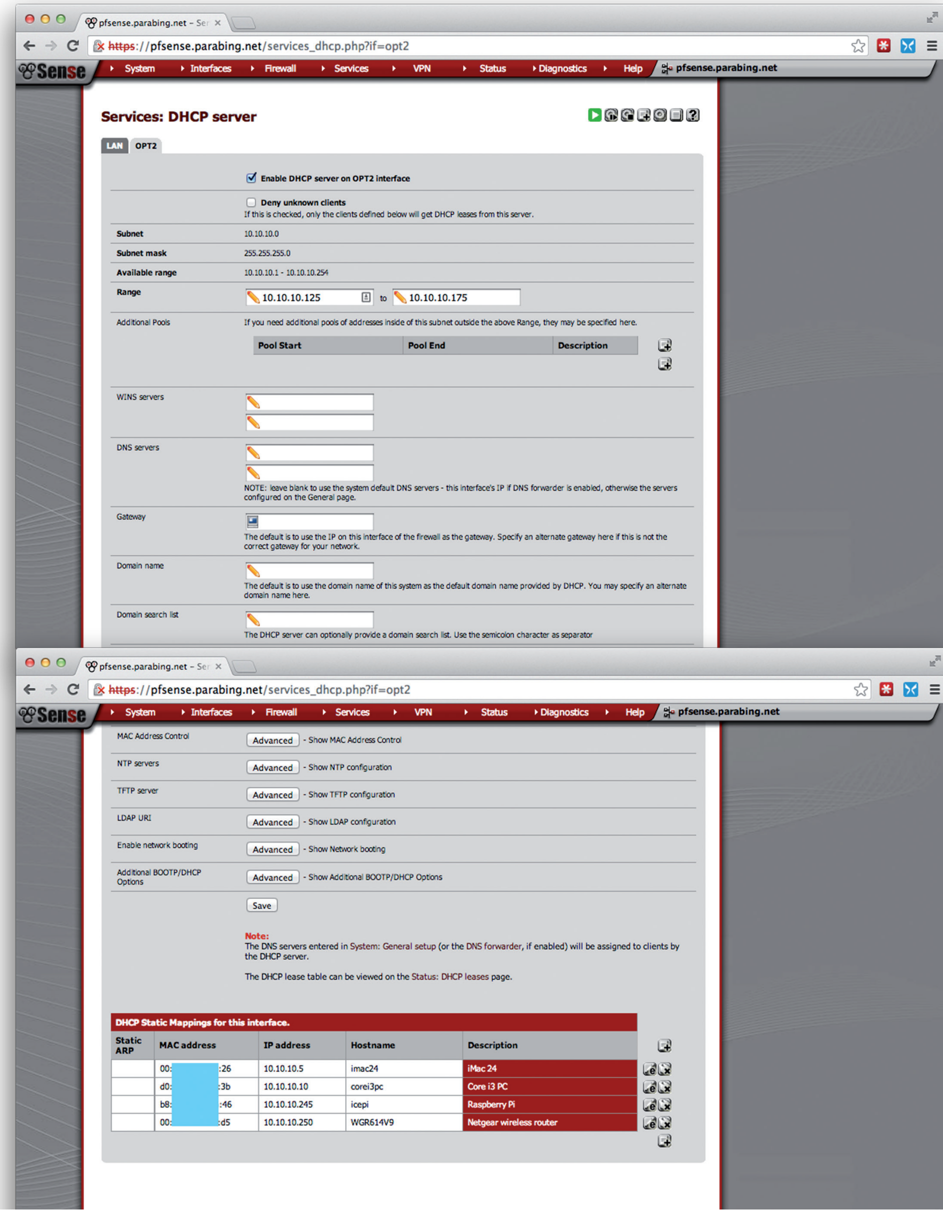
Εικόνα 3 (System → Advanced → Miscellaneous). Αξίζει να ενεργοποιήσετε κι εσείς τον power management daemon και να παίξετε με τις σχετικές ρυθμίσεις. Δεν φαίνεται από το screenshot, όμως αυτή τη στιγμή μάς έχει πιάσει μανία με την εξοικονόμηση ενέργειας. Ακριβώς γι' αυτό, στο "On AC Power Mode" έχουμε βάλει το προφίλ "Minimum", που σημαίνει ότι ο επεξεργαστής λειτουργεί όσο πιο συντηρητικά γίνεται. Να φανταστείτε, η μέγιστη ταχύτητα του δικού μας είναι 1,6GHz αλλά λόγω της σχετικής ρύθμισης δουλεύει στα 150MHz. Ενδιαφέρουσα είναι και η ρύθμιση "Thermal Sensors", ώστε όποτε είναι δυνατόν να βλέπουμε και τη θερμοκρασία που έχει ανεβάσει ο επεξεργαστής. Την περισσότερη ώρα, ο δικός μας είναι στους 28 ή 29 βαθμούς Κελσίου.



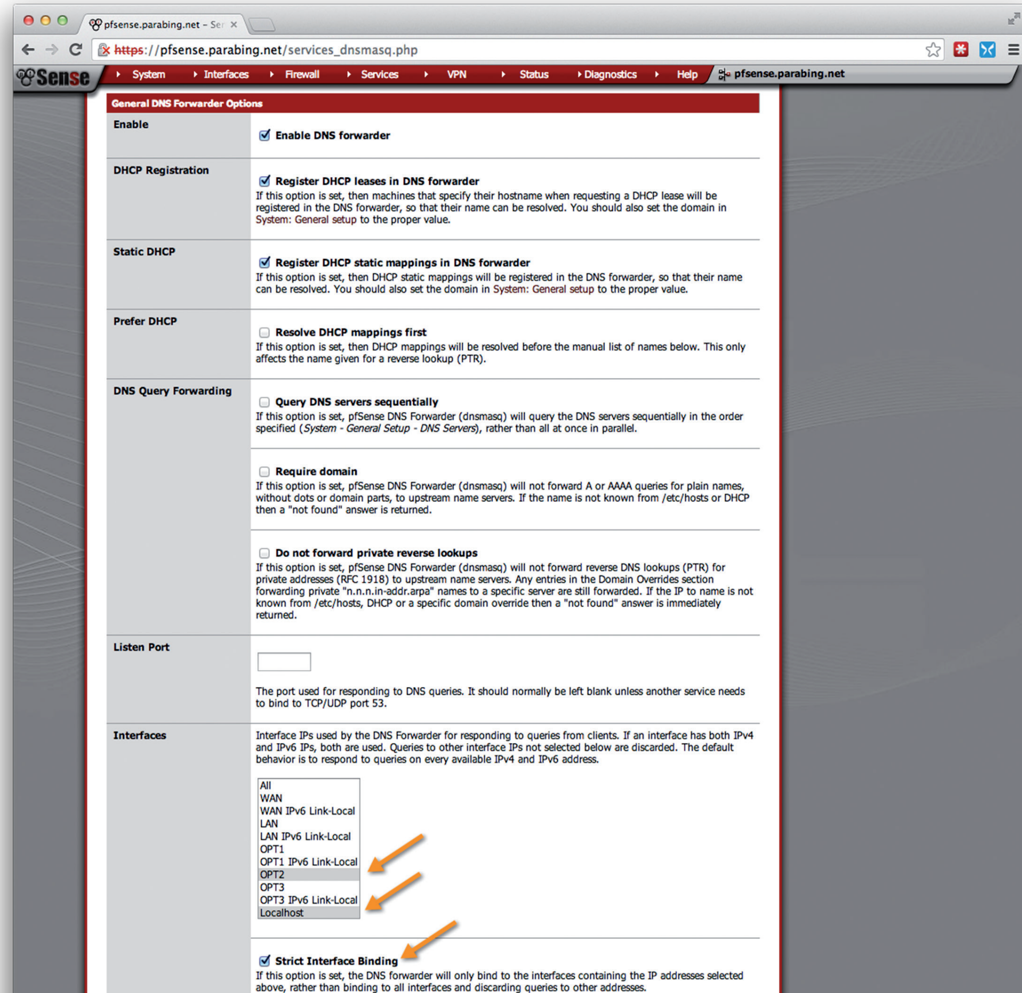
Εικόνα 4 (Interfaces → (assign)). Το box έχει δύο WAN interfaces: το WAN (re0) και το OPT3 (re2). Το WAN βλέπει ένα VDSL modem και το OPT3 ένα cable modem. Υπάρχει επίσης το LAN (em0) που είναι γεφυρωμένο με το OPT1 (re1) κι αμφότερα συγκροτούν ένα λογικό, bridge interface: το OPT2 (BRIDGE0). Από φυσικής σκοπιάς, στο Ethernet port του LAN είναι συνδεδεμένο ένα 1000άρι 3com switch τεσσάρων θυρών. Στο δε Ethernet port του OPT1 είναι συνδεδεμένο ένα Netgear WiFi router τύπου 802.11g, το οποίο, παρεμπιπτόντως, προσφέρει και τέσσερις 100άρες θύρες Ethernet που όμως δεν χρησιμοποιούνται. Η παρουσία του Netgear είναι πρόσκαιρη και υπάρχει μόνο και μόνο για να συνδέονται οι wireless clients (δύο laptops, δύο smartphones, ένα iPad κι ένα πολυμηχάνημα). Σύντομα το Netgear θα αντικατασταθεί από ένα "σκέτο" WiFi access point, τύπου 802.11n.



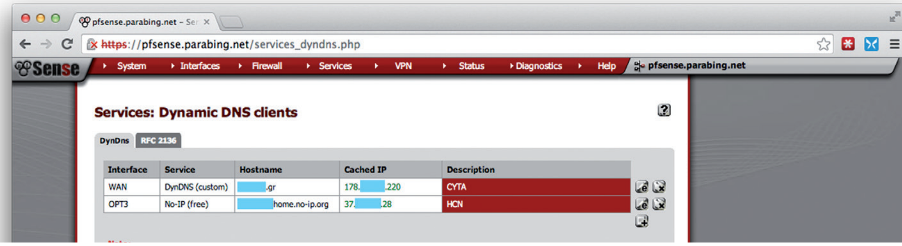
Εικόνα 5 (Interfaces → Bridges). Να και το μοναδικό bridge interface του box μας, το οποίο "ενώνει" λογικά τα LAN και OPT1 interfaces. Αν δεν ξέρετε τι ακριβώς είναι το bridging και σε τι χρησιμεύει, διαβάστε το άρθρο που αρχίζει από τη σελίδα 24 του deltaHacker 027. Στο δε άρθρο που ξεκινά από τη σελίδα 40 του ίδιου τεύχους, μπορείτε μεταξύ άλλων να δείτε πώς στήνουμε bridges στο pfSense.



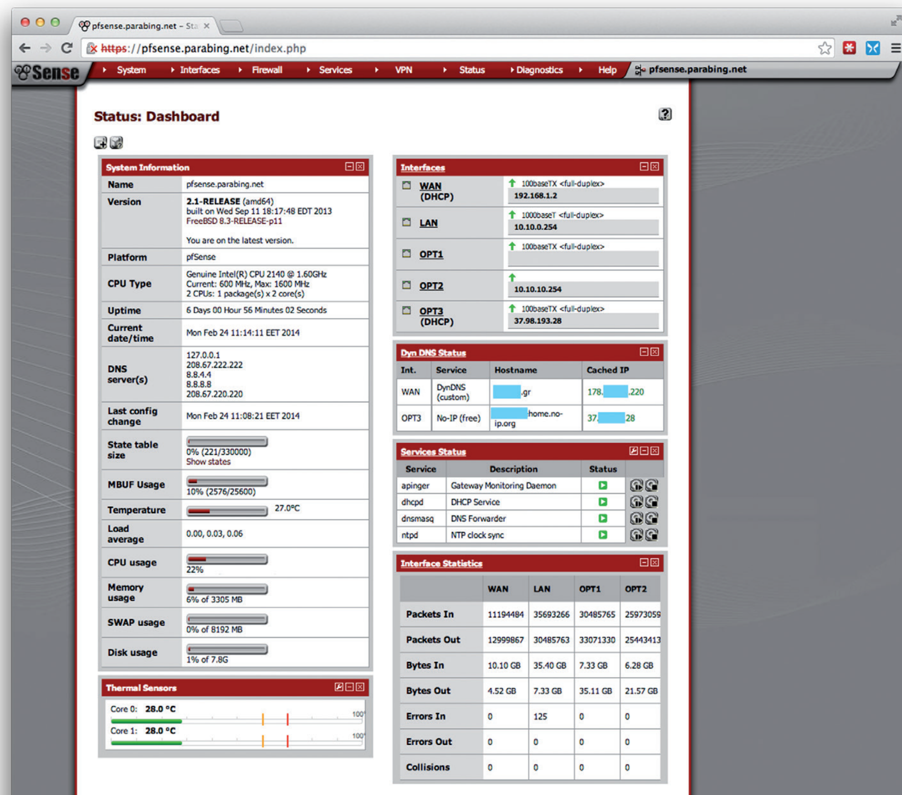
Εικόνα 6 (Services → DHCP server → OPT2). O DHCP server του pfSense box μοιράζει διευθύνσεις IP στους clients που "ακουμπάνε" στο OPT2 interface, το οποίο υπενθυμίζουμε ότι αποτελεί μια γέφυρα που περιλαμβάνει τα LAN και OPT1 interfaces. Δείτε εξάλλου ότι έχουμε φροντίσει ώστε σε συγκεκριμένα μηχανήματα/συσκευές ο DHCP server να δίνει πάντα την ίδια διεύθυνση IP.



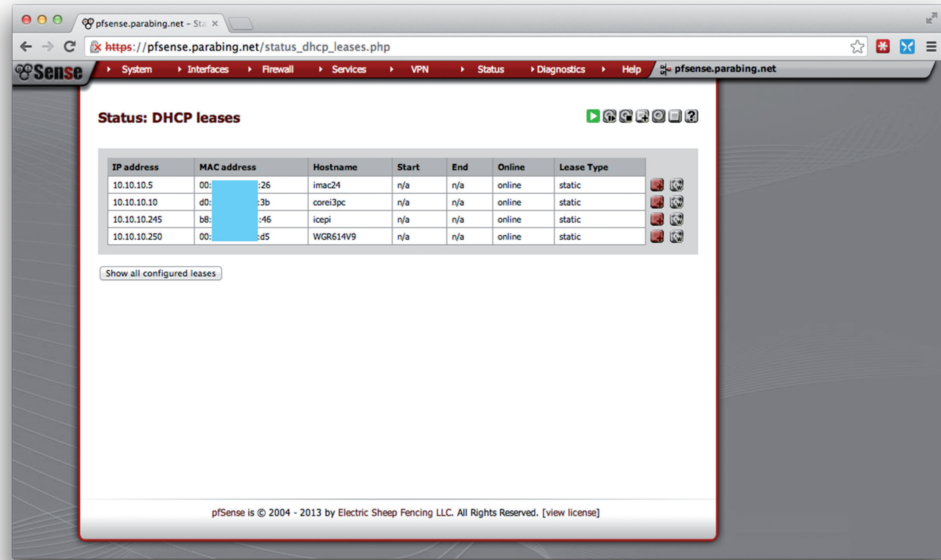
Εικόνα 7 (Services → DNS forwarder). Ο DNS forwarder του pfSense, μ' άλλα λόγια ο τοπικός μας nameserver, αφουγκράζεται για αιτήσεις πελατών αυστηρά από τα OPT2 και loopback (Localhost) interfaces. Αν δεν ακούει απ' αυτά τα interfaces ή/και δεν είναι τσεκαρισμένο το Strict Interface Binding, τότε το bridging δεν θα δουλεύει σωστά, ούτε και το load balancing (βλ. συνέχεια).



Εικόνα 8 (Services → Dynamic DNS). Χρησιμοποιούμε δύο διαφορετικές υπηρεσίες dynamic DNS (Dyn και No-IP), ώστε να φτάνουμε στο pfSense box όταν είμαστε εκτός τοπικού δικτύου *χωρίς* να γνωρίζουμε καθένα από τα δύο δυναμικά IPs των WAN interfaces του router μας. Κανονικά, σ' αυτό το screenshot θα έπρεπε να είναι "μουτζουρωμένα" σχεδόν τα πάντα. Αλλά έχουμε λόγους ώστε να μην ανησυχούμε πολύ για αδιάκριτα scannings ή/και επιθέσεις. (Ό,τι γράψαμε μόλις, πραγματικά ελπίζουμε να μην αποδειχθεί ότι ταιριάζει στην κατηγορία "Famous Last Words".)



Εικόνα 9 (Status → Dashboard). Το ωραιότατο dashboard του δικού μας super-router. Να φτιάξετε κι εσείς έναν — δεν θα το μετανιώσετε!



Εικόνα 10 (Status → DHCP Leases). Όλοι οι υπολογιστές (και το ένα wireless router της Netgear) που αυτή τη στιγμή εξυπηρετούνται από το pfSense, έχουμε φροντίσει ώστε να παίρνουν πάντα την ίδια διεύθυνση IP. Σημειώστε ότι στα μηχανήματα πίσω από το pfSense δεν είναι απαραίτητο ν' αναφερόμαστε με βάση το IP τους: το hostname αρκεί.

Υπάρχουν διάφορες όψεις του web panel που σκοπίμως αποφύγαμε. Θα παρατηρήσατε, για παράδειγμα, ότι δεν πήγαμε καν προς τα μέρη του μενού ονόματι Firewall. Πάντως σε πολύ λίγο, που θα καταπιαστούμε με το multi-WAN, να είστε σίγουροι ότι θα έχουμε πολλούς λόγους για να πειράξουμε τους κανόνες του PF — ευτυχώς πάντα μέσα από το εύχρηστο web panel του pfSense.

Υποστήριξη multi-WAN

Προφανώς, το να έχουμε δύο κάρτες δικτύου όπου η κάθε μία έχει μπροστά της ένα διαφορετικό modem, δεν είναι αρκετό ώστε το pfSense να χρησιμοποιεί και τις δύο συνδέσεις προς το Internet. Για να επιτευχθεί κάτι τέτοιο, οφείλουμε:

- να βεβαιωθούμε ότι τουλάχιστον ένας nameserver αντιστοιχεί σε κάθε ένα από τα WAN gateways.
- να ορίσουμε ένα gateway group, το οποίο θα περιλαμβάνει τα δύο διαφορετικά WAN gateways
- να εμπλέξουμε το συγκεκριμένο gateway group σε κατάλληλο κανόνα του firewall για το LAN interface (ή για το όποιο bridge interface έχουμε ρυθμισμένο)

Ακούγεται απλή και λογική η όλη διαδικασία — και πράγματι είναι. Κρύβει βεβαίως κάποιες λεπτομέρειες, τις οποίες οφείλουμε να γνωρίζουμε. Για να μη θεωρητικολο-

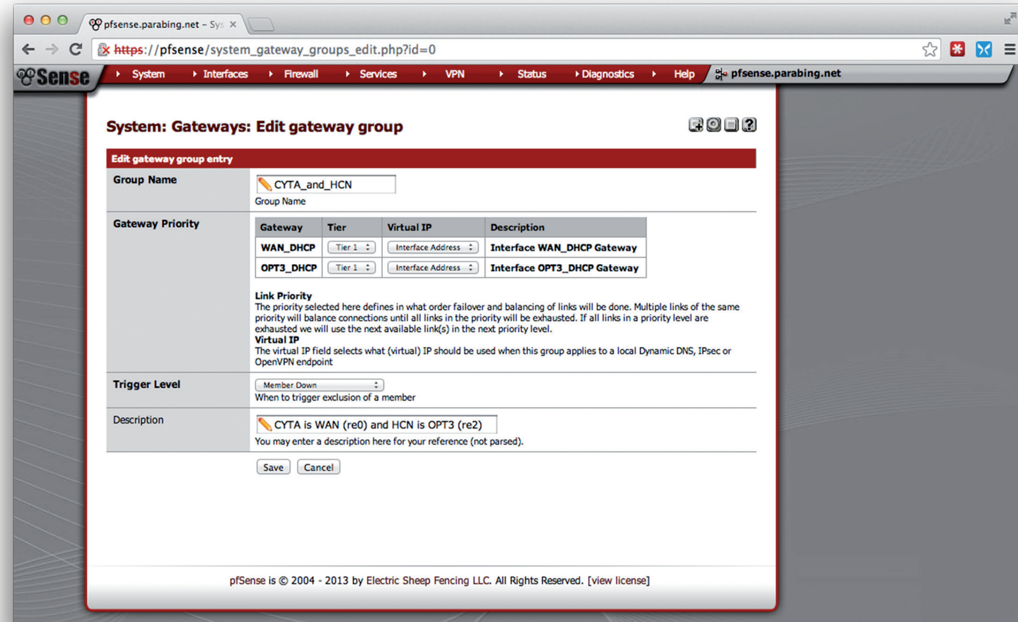
γούμε αλλά και για ν' αποφύγουμε το πελάγωμα, θα περιγράψουμε στη συνέχεια το δικό μας multi-WAN setup, για δύο διαφορετικές συνδέσεις. Παρόμοιο θα είναι και το δικό σας setup, ακόμη κι αν έχετε περισσότερες από δύο ιντερνετικές συνδέσεις.

Στο δικό μας box, λοιπόν, τα δύο network interfaces που βλέπουν τα δύο modem είναι τα WAN (default gateway) και OPT3 (βλ. Interfaces → (assign)). Αμφότερα είναι ενεργοποιημένα και παίρνουν διεύθυνση IP από το αντίστοιχο modem/router, δυναμικά. Δυναμικά παίρνουν και τη διεύθυνση του gateway. Σε καθένα από τα gateways έχουμε αντιστοιχίσει δύο διαφορετικούς, δημόσιους nameservers (βλ. System → General Setup → DNS servers). Προσοχή: Αν κάποιο από τα WAN ή OPT3 χρησιμοποιούσε στατικό IP, τότε θα έπρεπε να του έχουμε καθορίσει την κατάλληλη διεύθυνση gateway (π.χ., Interfaces → (assign) → WAN → Static IPv4 configuration → Gateway).

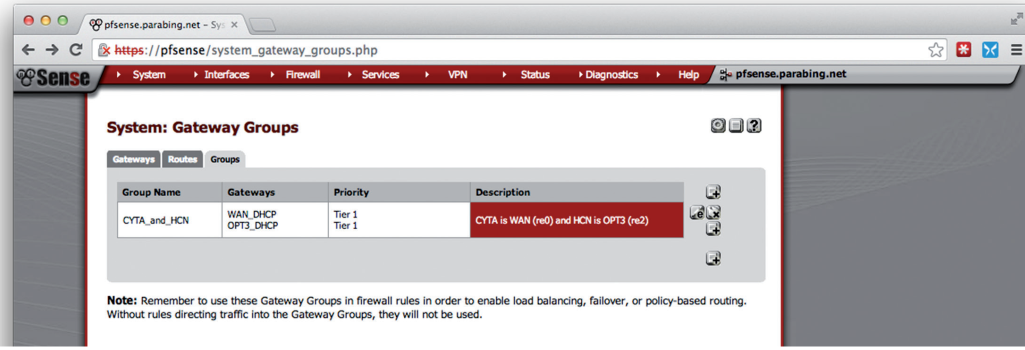
Προκειμένου να ορίσουμε gateway group που θα περιλαμβάνει τα δύο WAN gateways που μας ενδιαφέρουν (ένα για κάθε σύνδεση προς το Internet), πηγαίνουμε στο System → Routing → Groups. Εκεί, με κλικ στο [+] δημιουργούμε ένα νέο gateway group. Δείτε στην εικόνα 11 το δικό μας και διαβάστε την αντίστοιχη περιγραφή. Στην εικόνα 12 βλέπετε το ένα και μοναδικό gateway group που έχουμε φτιάξει, ενώ στη 13 φαίνονται τα δύο gateways του box μας. Καθένα εξ αυτών συμμετέχει στο gateway group με το ίδιο βάρος (weight), που σημαίνει ότι χρησιμοποιείται όσες φορές και το άλλο. Με άλλα λόγια, όπως είναι τώρα τα gateways χρησιμοποιούνται εναλλάξ για συνδέσεις προς μηχανήματα ή/και υπηρεσίες *εκτός* τοπικού δικτύου[6]. Αν κάνουμε κλικ στο [e], δίπλα από ένα gateway, διαπιστώνουμε ότι μπορούμε ν' αλλάξουμε το βάρος του (κάντε κλικ και στο κουμπί Advanced — βλ. εικόνα 14). Γιατί όμως να το θέλαμε αυτό; Σκεφτείτε το εξής, απλό σενάριο: Οι δύο ιντερνετικές συνδέσεις δεν έχουν την ίδια ταχύτητα και η μία, π.χ., είναι δύο φορές πιο γρήγορη από την άλλη (μιλάμε τώρα για τις πραγματικές ταχύτητες κι όχι για τις ονομαστικές). Σ' αυτή την περίπτωση, η αργή γραμμή θα φορτώνεται άσκοπα τη στιγμή που η άλλη, η γρήγορη, θα παραμένει κατά βάση ανεκμετάλλευτη. Πώς βελτιώνουμε αυτή την κατάσταση; Πολύ απλά, στο gateway της γρήγορης γραμμής δίνουμε βάρος 2 και στο gateway της αργής δίνουμε βάρος 1.

Όσο παράξενο κι αν φαίνεται, το pfSense είναι πλέον πανέτοιμο για να χρησιμοποιεί και τις δύο συνδέσεις προς το Internet. Αν όμως δεν ορίσουμε τον κατάλληλο κανόνα στο firewall για το LAN (για εμάς το OPT2, λόγω bridging), τότε θα συνεχίσει να χρησιμοποιεί τη σύνδεση του default gateway. Δίνουμε λοιπόν Firewall → Rules → LAN (εμείς OPT2) κι ορίζουμε ένα νέο κανόνα που λέει, στην ουσία, αυτό: "οποιαδήποτε σύνδεση ξεκινά από το τοπικό δίκτυο, δρομολόγησέ τη να χαρείς μέσω του gateway group" (κι όχι μέσω ενός εκ των δύο gateways που έχει το box — βλ. εικόνα 15). Εκτός από τον κανόνα που μόλις ορίσαμε, εξαιτίας του multi-WAN κι αναλόγως του τι συμβαίνει στο τοπικό μας δίκτυο, είναι πιθανό να χρειαστούν πρόσθετοι κανόνες για το firewall ή/και κάποιοι κανόνες για το port forwarding. Δείτε τις εικόνες 16 και 17 κι οπωσδήποτε διαβάστε τις αντίστοιχες περιγραφές.

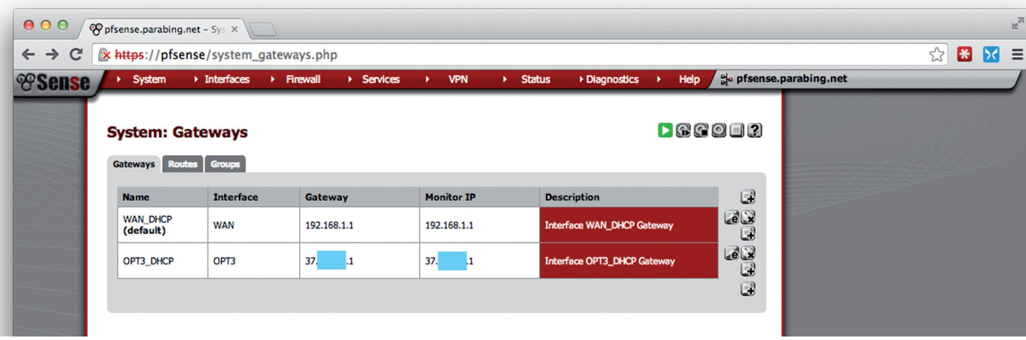
6. Υπάρχει προφανώς μια αμφισημία με τον όρο "σύνδεση", όπου ανάλογα με τα συμφραζόμενα αναφέρεται σε φυσική σύνδεση (π.χ., μιας κάρτας Ethernet με το modem) ή σε λογική σύνδεση (π.χ., από έναν browser προς έναν web server).



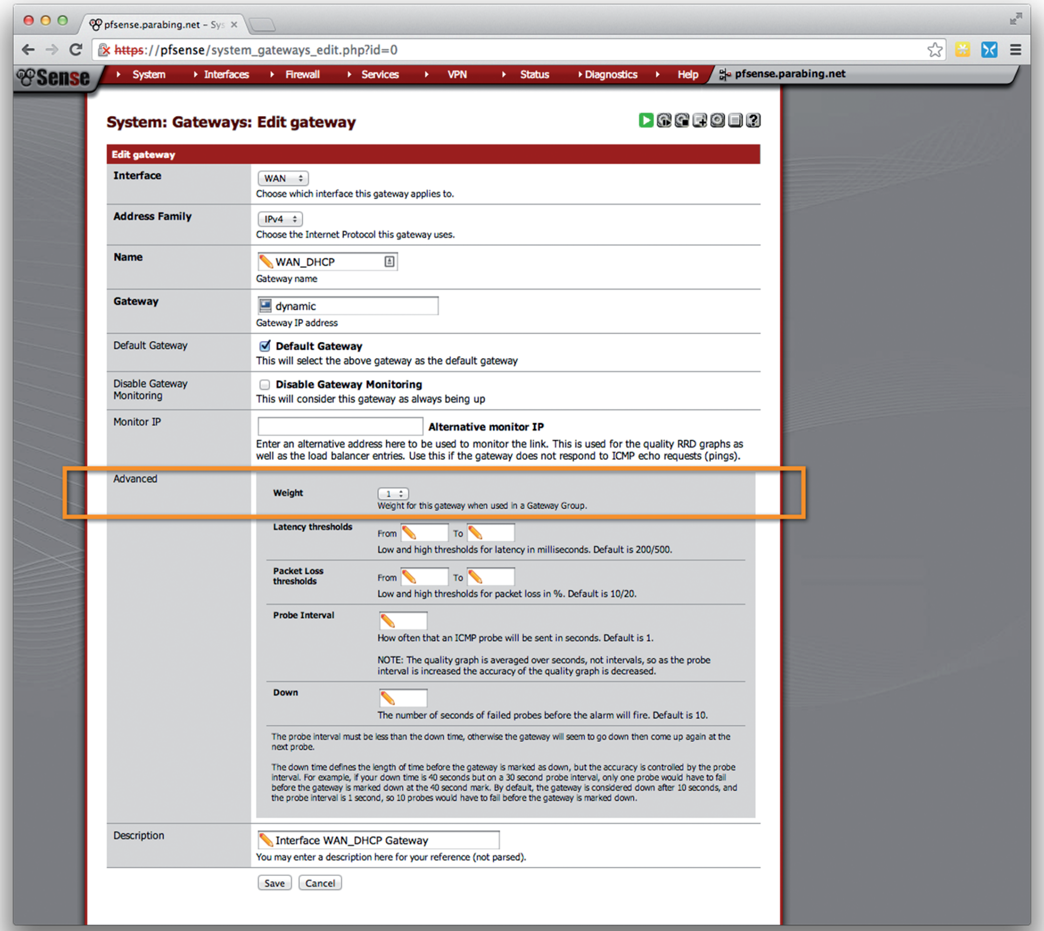
Εικόνα 11 (System → Routing → Groups → [e]). Ιδού τα δύο gateways του gateway group μας, ένα για κάθε μία από τις διαθέσιμες συνδέσεις στο Internet. Αμφότερα βρίσκονται στο ίδιο επίπεδο (tier), που σημαίνει ότι προτιμούνται το ίδιο κι έτσι το pfSense κάνει load balancing μεταξύ τους. Για διαδοχικές συνδέσεις, λοιπόν, χρησιμοποιούνται εναλλάξ. Αν τα δύο gateways δεν ανήκαν στο ίδιο tier, τότε το pfSense θα έκανε failover. Για την ακρίβεια, θα χρησιμοποιούσε διαρκώς το gateway του χαμηλότερου tier κι αν αυτό έπεφτε θα στρεφόταν στο άλλο, του υψηλότερου tier. (Σε σενάρια failover οι gateways των χαμηλότερων tiers προτιμούνται από αυτούς των υψηλότερων.) Παρατηρήστε εξάλλου ότι υπάρχουν τέσσερα συμβάντα (triggers) που είναι δυνατόν να προκαλέσουν την αφαίρεση ενός gateway από το tier στο οποίο ανήκει: Member Down (το προκαθορισμένο συμβάν), Packet Loss, High Latency και Packet Loss or High Latency. Κατά τη διάρκεια των δοκιμών μας βιώσαμε τα αποτελέσματα του πρώτου συμβάντος — κι όχι επειδή κάναμε κάποια εξομοίωση: Η γραμμή VDSL είχε πέσει και το αντιληφθήκαμε από ένα download που έμοιαζε να αργεί περισσότερο απ' όσο έπρεπε. (Για την ιστορία, ας αναφέρουμε ότι προκειμένου να ενεργοποιηθεί ξανά η VDSL χρειάστηκε να τηλεφωνήσουμε στον ISP και να κάνουν reset της σύνδεσης από τη μεριά τους. Κατά πώς φαίνεται, το πρόβλημα προκλήθηκε για άλλη μια φορά από την κακή κατάσταση της καλωδίωσης έξω από την πολυκατοικία — ή τουλάχιστον αυτή είναι η επίσημη εξήγηση που για άλλη μια φορά πήραμε.)



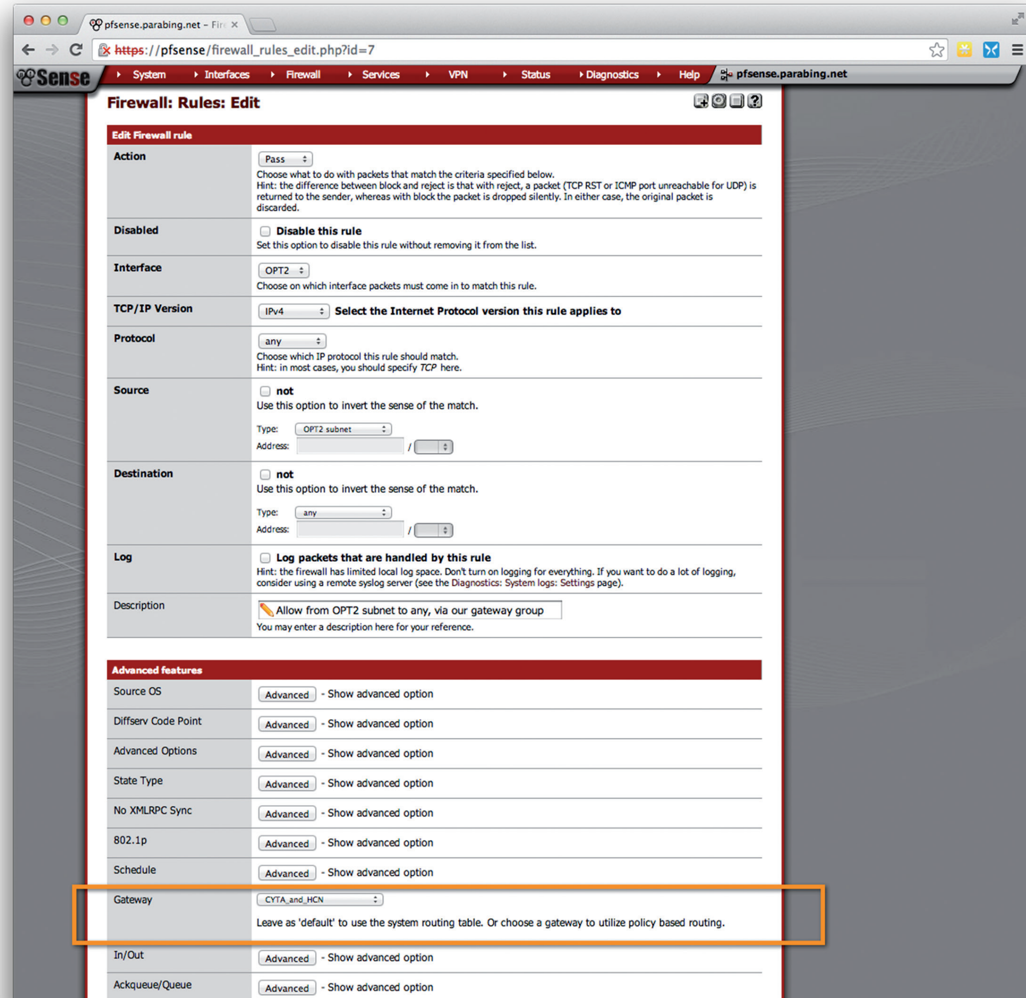
Εικόνα 12 (System → Routing → Groups). Το ένα και μοναδικό gateway group που έχουμε ορίσει και χρησιμοποιούμε, με τα μέλη του και την αναλυτική του περιγραφή. Σημειώστε ότι επιτρέπεται να έχουμε περισσότερα από ένα gateway groups, τα οποία θα εμπλέκουμε σε διαφορετικούς κανόνες του firewall. Παράδειγμα: Ενώ γενικά οι πελάτες του OPT2 interface βγαίνουν στον έξω κόσμο μέσω ενός failover gateway group, ένα συγκεκριμένο μηχάνημα (το λεγόμενο "κατεβαστήρι";) βγαίνει στο Internet μέσω ενός load balancing gateway group.



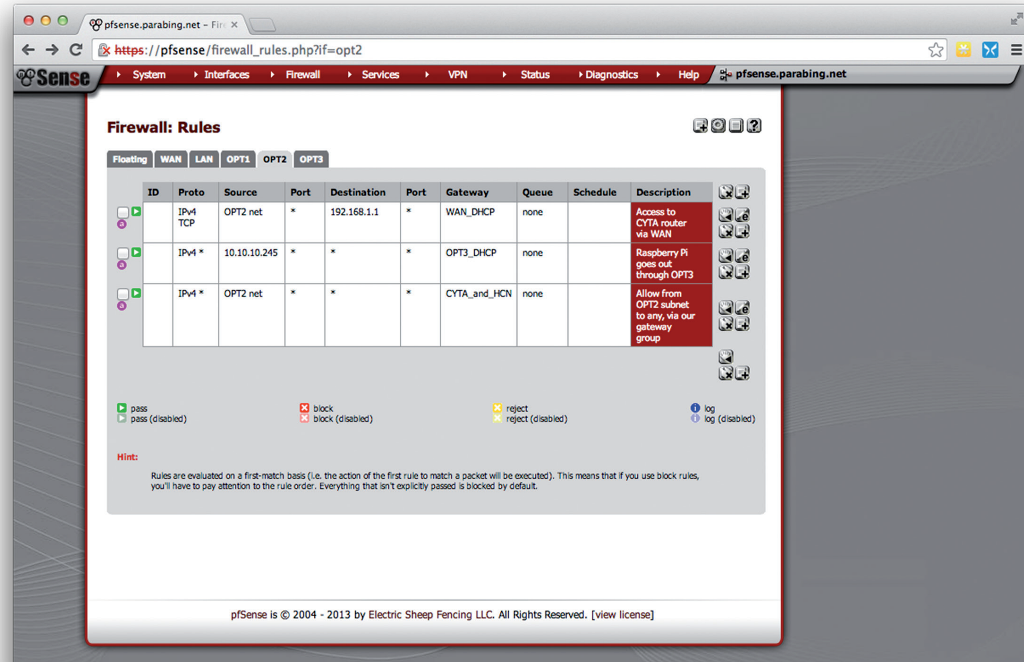
Εικόνα 13 (System → Routing → Gateways). Για κάθε gateway υπάρχει ένα Monitor IP. Πρόκειται για μία διεύθυνση στην οποία το pfSense κάνει ping, προκειμένου να ελέγχει αν η αντίστοιχη ιντερνετική σύνδεση λειτουργεί. Όποτε διαπιστώνει ότι δεν λειτουργεί, τότε η σύνδεση θεωρείται ανενεργή κι αν ανήκει σε κάποιο gateway group εννοείται πως παύει να χρησιμοποιείται. Εξ ορισμού, η διεύθυνση στην οποία στέλνονται τα ping packets είναι εκείνη του ίδιου του gateway. Ενδέχεται όμως τα pings να φτάνουν στον gateway, ωστόσο το link προς τον ISP να έχει πέσει. Μια εναλλακτική προσέγγιση, λοιπόν, είναι ως Monitor IP να βάζουμε τη διεύθυνση ενός δημόσιου, ιντερνετικού server, ο οποίος φυσικά απαντά στα ping requests. Αλλά και πάλι, η επιλογή αυτού του server πρέπει να γίνεται με προσοχή: Αν πέσει, το pfSense θα θεωρήσει ότι το αντίστοιχο link έχει επίσης πέσει ενώ στην πραγματικότητα αυτό μάλλον θα 'ναι μια χαρά. Δύο δημόσιες, αξιόπιστες διευθύνσεις IP στις οποίες γίνονται αποδεκτά τα pings, είναι εκείνες των δύο δημοφιλών nameservers της Google: 8.8.8.8 και 8.8.4.4.



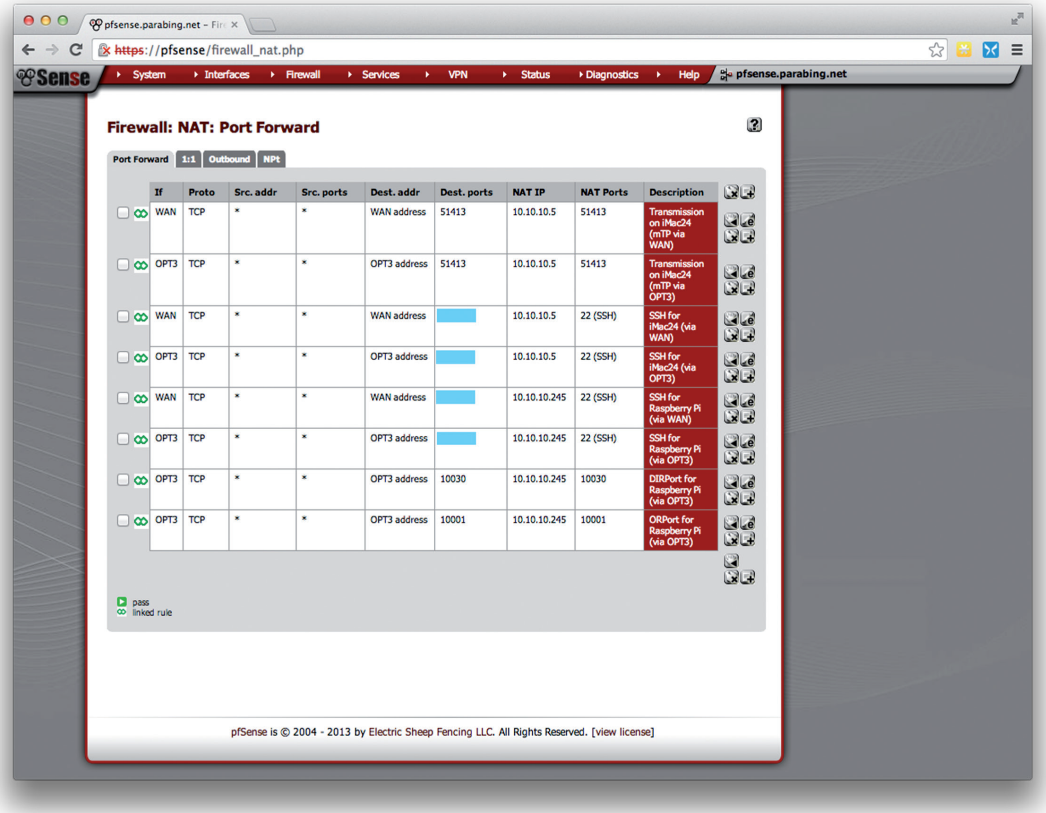
Εικόνα 14 (System → Routing → Gateways → ... → [e]). Η ανάθεση διαφορετικών βαρών (weights) στα gateways έχει νόημα όταν, π.χ., οι αντίστοιχες γραμμές διαφέρουν σημαντικά ως προς την ταχύτητα αλλά συμμετέχουν στο ίδιο load balancing gateway group. Έτσι, αν η μία γραμμή είναι τρεις φορές πιο γρήγορη από την άλλη κι αμφότερες ανήκουν στο ίδιο gateway group, τότε στη γρήγορη αξίζει να αναθέσουμε βάρος 3 και στην αργή βάρος 1. Κατ' αυτόν τον τρόπο οι γραμμές θα "φορτώνονται" αναλόγως της χωρητικότητάς τους και δεν θα έχουμε καθυστερήσεις επειδή η μία μένει ανεκμετάλλευτη τη στιγμή που η άλλη δεν μπορεί να ανταπεξέλθει.



Εικόνα 15 (Firewall → Rules → LAN (ή OPT-κάτι) → ... [e]). Θέλουμε όλα τα μηχανήματα πίσω από το bridge interface του pfSense box να χρησιμοποιούν και τις δύο διαθέσιμες δυνδέσεις στο Internet. Για να το πετύχουμε, αρκεί να τροποποιήσουμε ελαφρώς τον κανόνα για το OPT2 που λέει: "άφηνε ελεύθερα τα πακέτα IPv4, ανεξαρτήτως πρωτοκόλλου, αρκεί να προέρχονται πίσω από το OPT2". Ο νέος κανόνας που προκύπτει έχει μία μόνο διαφορά σε σχέση με τον προηγούμενο: Για τα πακέτα που βγαίνουν στο Internet δεν χρησιμοποιείται το προκαθορισμένο gateway ή κάποιο άλλο, αλλά το gateway group που μας ενδιαφέρει. Όλα τα υπόλοιπα, δηλαδή οι λεπτομέρειες του load balancing ή του failover, διευθετούνται αυτόματα. Αν στο δικό σας setup δεν έχετε κάνει bridging, τότε ο κανόνας αυτός θα αφορά στο LAN interface.



Εικόνα 16. Εξαιτίας του load balancing, για κάποια μηχανήματα του τοπικού σας δικτύου ενδέχεται να προσθέσετε μερικούς νέους κανόνες στο firewall προκειμένου συγκεκριμένες υπηρεσίες να λειτουργούν απροβλημάτιστα. Εμείς, π.χ., το Raspberry Pi το χρησιμοποιούμε ως Tor relay (βλ. και http://deltahacker.gr/raspi-tor-relay). Δεν θέλουμε ν' αλλάζει διαρκώς το δημόσιο IP του στο δίκτυο του Tor, οπότε ορίσαμε έναν κανόνα που φροντίζει ώστε η δικτυακή κίνηση του Pi να βγαίνει προς τα έξω πάντα μέσω του ίδιου gateway. Μπροστά από το WAN interface, εξάλλου, βρίσκεται το modem του VDSL provider που χρησιμοποιούμε. (Η συσκευή, παρεμπιπτόντως, δεν παρέχει τη δυνατότητα απενεργοποίησης του routing, οπότε τη διευθυνση IP που παίρνει ο μοναδικός του πελάτης, που είναι το pfSense μέσω του WAN interface, τη βάλαμε στο DMZ.) Επειδή μερικές φορές θέλουμε να συνδεόμαστε στο web panel του VDSL modem/router, ορίσαμε έναν κανόνα στο firewall του pfSense ο οποίος όποτε βλέπει πακέτα από το OPT2 subnet να πηγαίνουν προς το 192.168.1.1 (αυτή είναι η διεύθυνση του LAN interface που έχει το modem/router), τα στέλνει προς τα εκεί μέσω του WAN gateway. (Μέσω του OPT3 gateway δεν θα έφταναν ποτέ!)



Εικόνα 17 (Firewall → NAT → Port Forward). Οι κανόνες για port forwarding δεν θα μπορούσαν να λείπουν από το box μας. Όταν, π.χ., είμαστε εκτός τοπικού δικτύου, τότε θέλουμε να φτάνουμε στο Raspberry Pi (10.10.10.245) μέσω SSH — και θέλουμε να το κάνουμε από οποιαδήποτε από τις δύο συνδέσεις προς το Internet που διαθέτει το pfSense. Γι' αυτό το λόγο δεν προσθέτουμε έναν αλλά δύο κανόνες port forwarding, οι οποίοι διαφέρουν ως προς το destination address. Το ίδιο θέλουμε να συμβαίνει και με τον iMac (10.10.10.5), οπότε προσθέτουμε άλλους δύο σχετικούς κανόνες. Όπως ήδη αναφέραμε, εξάλλου, το Raspberry Pi λειτουργεί κι ως Tor relay (μέσω της σύνδεσης WAN), επομένως τα DIRPort και ORPort του Pi θέλουμε να 'ναι προσβάσιμα μέσω της δημόσιας διεύθυνσης του modem/router *μπροστά* από το WAN interface. Πώς το διασφαλίζουμε αυτό; Μα, με άλλους δύο κανόνες port forwarding. Στον iMac, τέλος, μερικές φορές τρέχουμε τον BitTorrent client ονόματι Transmission. Για αποτελεσματικότερη διακίνηση δεδομένων μέσω του Transmission με ταυτόχρονη αποφυγή προβλημάτων σε άλλες δικτυακές εφαρμογές, έχουμε ενεργοποιήσει την υποστήριξη του πρωτοκόλλου μTP. Το port που έχουμε καθορίσει για το μTP είναι το 51413/TCP και φυσικά έχουμε δημιουργήσει δύο σχετικούς κανόνες port forwarding, έναν για κάθε WAN interface του pfSense.

Δοκιμές κι εντυπώσεις

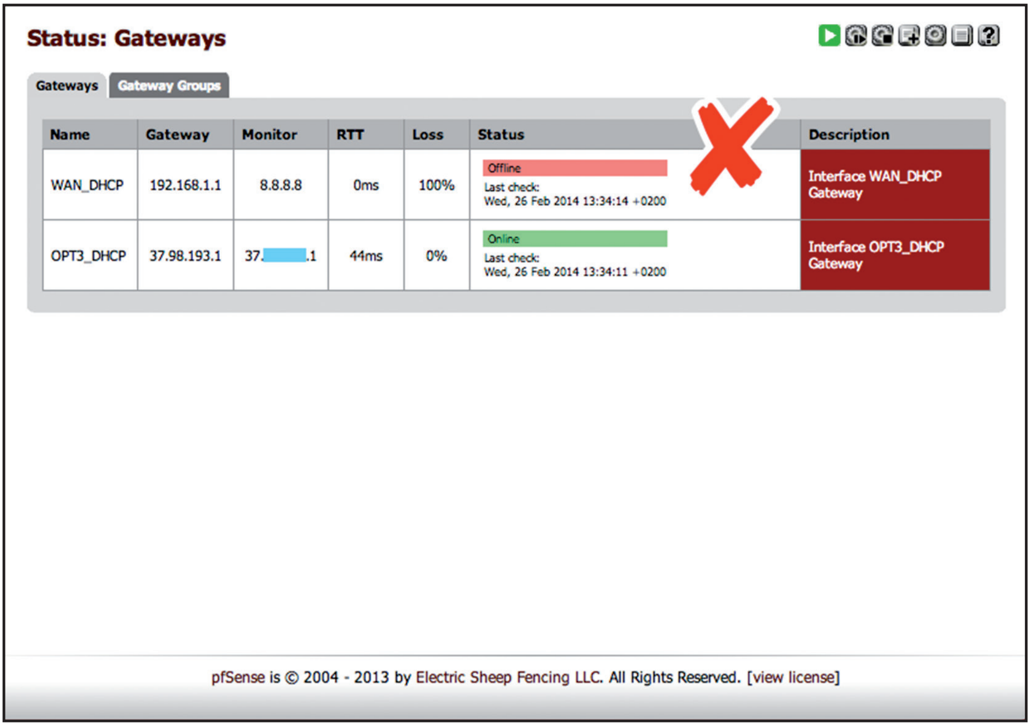
Δεν έχουμε λόγια για να σας περιγράψουμε πώς βλέπουμε τη μετά-το-pfSense-με-load-balancing εποχή. Ή μάλλον, κάτι μπορούμε να κάνουμε: Πριν στήσουμε το pfSense box, το τοπικό μας δίκτυο ήταν μισερό. Πλέον, έχει αναβαθμιστεί στην κατηγορία Kick-Ass (TM). Εντάξει, υπερβάλλουμε λίγο. Υπάρχουν πολλά πράγματα που έχουμε να βελτιώσουμε εδώ γύρω —πάντα όσον αφορά στη δικτυακή υποδομή—, ωστόσο η παρουσία ενός ικανότατου κι ασφαλέστατου router, ο οποίος μπορεί να τρέχει εξαιρετικά ενδιαφέρουσες εφαρμογές και παράλληλα συνδυάζει δύο ιντερνετικές συνδέσεις, είναι πράγματι καταλυτική. Γι' αυτές τις έξτρα εφαρμογές, παρεμπιπτόντως, θα μιλήσουμε σε επόμενα άρθρα κι επεισόδια του deltaCast. Προς το παρόν θα σας αφήσουμε με δύο παρατηρήσεις.

Η πρώτη αφορά σε κάποια προβλήματα που ενδέχεται να δημιουργήσει το load balancing, κυρίως με τα συστήματα web banking ορισμένων τραπεζών: Αν κατά τη διάρκεια ενός session δουν requests από διαφορετικά IPs, τότε αμέσως θα διακόψουν τη σύνδεση. Σε περιπτώσεις σαν αυτή, μπορούμε να ορίσουμε στο firewall έναν κανόνα για τον "ευαίσθητο" προορισμό (το web site της τραπεζας), ο οποίος θα δρομολογεί το traffic μέσω συγκεκριμένου gateway. Αν βέβαια είστε σαν εμάς, τότε η ιδέα τέτοιων κανόνων στο ruleset του firewall δεν θα σας ενθουσιάζει. Μια εναλλακτική προσέγγιση, λοιπόν, είναι να συνδέεστε σε έναν απομακρυσμένο OpenVPN server[7], να κάνετε το web banking σας και μετά να αποσυνδέεστε.

Η δεύτερη παρατήρηση έχει να κάνει με την ταχύτητα, όπως την αντιλαμβανόμαστε λόγω multi-WAN με load balancing. Κατά το τυπικό, χαλαρό web surfing, η αλήθεια είναι ότι αν στο τοπικό δίκτυο δεν υπάρχουν πολλοί χρήστες, τότε δύσκολα θα καταλάβουμε την όποια βελτίωση. Εκεί όμως που το load balancing πραγματικά λάμπει —μάλιστα ακόμη κι αν στο τοπικό δίκτυο υπάρχει ένας μόνο χρήστης— είναι κατά τη διάρκεια του downloading με πολλές, ταυτόχρονες συνδέσεις. Και φυσικά, τέτοιου είδους downloading έχουμε όταν κατεβάζουμε μέσω BitTorrent. Όπως κι εσείς θα μπορέσετε να διαπιστώσετε, η πραγματική ταχύτητα download ισούται, πράγματι, με το άθροισμα των ταχυτήτων των επί μέρους γραμμών.

Τι τα θέλετε; pfSense FTW — and then some!

7. Τρέχετε OpenVPN στο VPS σας, έτσι δεν είναι; :P



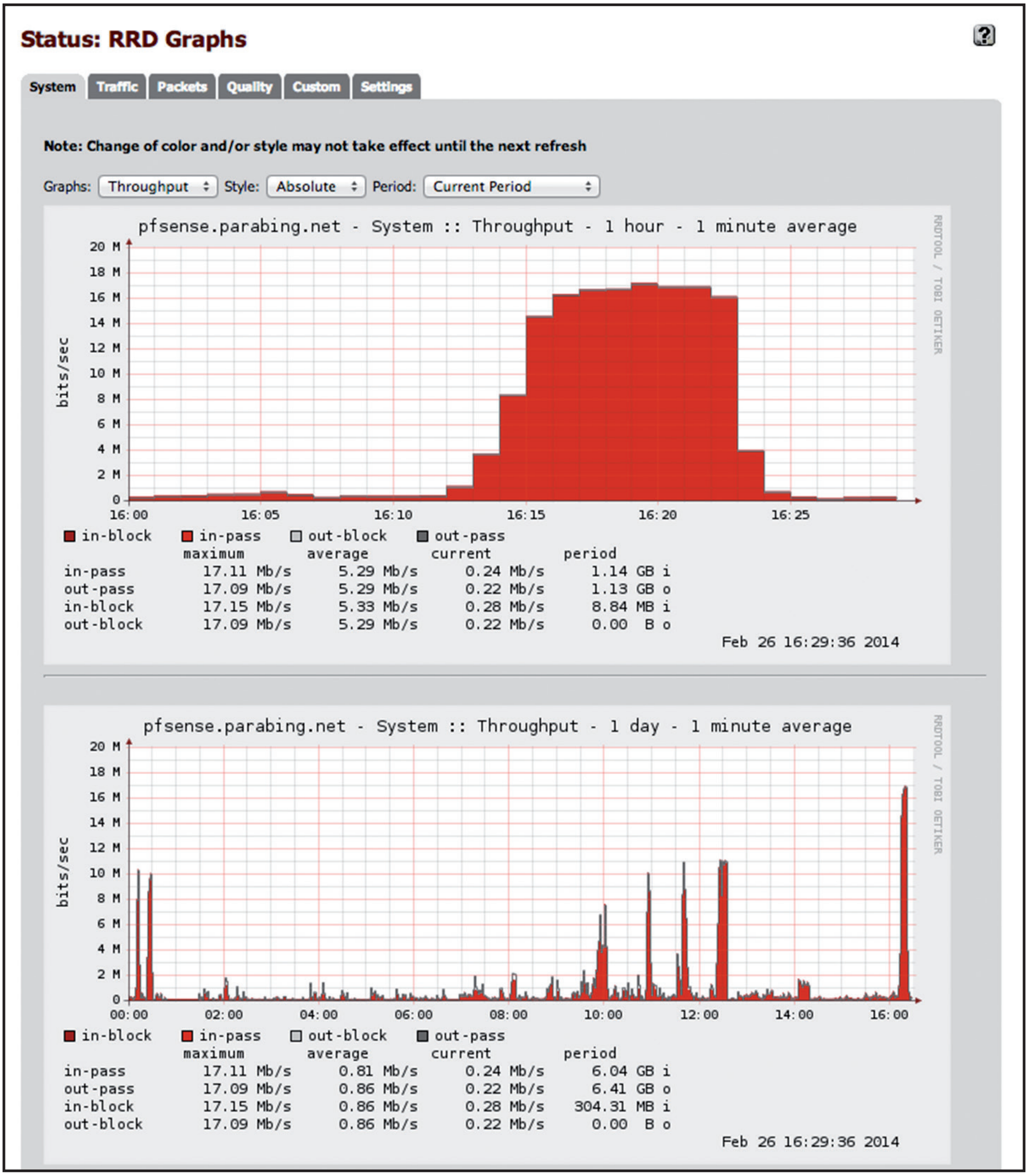
Status: Gateways

Gateways | Gateway Groups

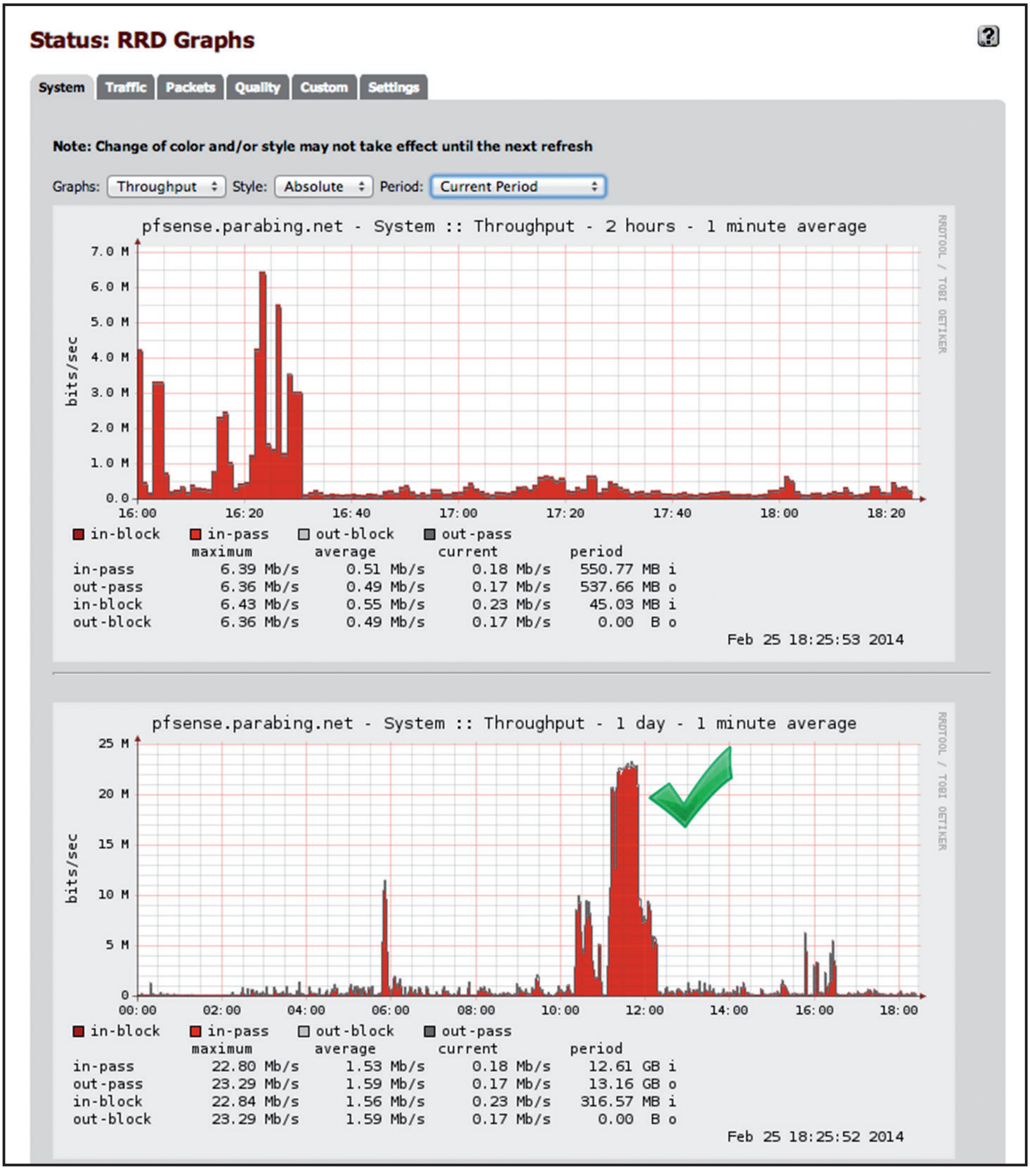
Name	Gateway	Monitor	RTT	Loss	Status	Description
WAN_DHCP	192.168.1.1	8.8.8.8	0ms	100%	Offline Last check: Wed, 26 Feb 2014 13:34:14 +0200	Interface WAN_DHCP Gateway
OPT3_DHCP	37.98.193.1	37.[redacted].1	44ms	0%	Online Last check: Wed, 26 Feb 2014 13:34:11 +0200	Interface OPT3_DHCP Gateway

pfSense is © 2004 - 2013 by Electric Sheep Fencing LLC. All Rights Reserved. [view license]

Πριν φτιάξουμε το pfSense box με το θαυμάσιο load balancing, η αποτυχία μίας από τις δύο συνδέσεις μας στο Internet θα προκαλούσε μίνι κρίση στο μικρόκοσμό μας — ενδεχομένως και μίνι, σύντομη κατάθλιψη. Κατά την προετοιμασία του παρόντος άρθρου, η σύνδεση VDSL πράγματι έπεσε. Μάλιστα έμεινε κάτω για κανένα 6ωρο, μέχρι να βγάλουμε άκρη με τον ISP (κακή ποιότητα καλωδίων στην πολυκατοικία, μην τα ρωτάτε). Το pfSense, πάντως, δεν πολυσκοτίστηκε: Μάρκαρε τη σχετική σύνδεση ως offline και συνέχισε να δρομολογεί τα πακέτα μέσω του gateway της άλλης σύνδεσης, που έχουμε από τον cable provider. Κανένα πρόβλημα, λοιπόν. Μη σας πούμε ότι χαρήκαμε και λίγο, αφού είχαμε την ευκαιρία να δοκιμάσουμε το load balancing σε αληθινό σενάριο κρίσης!



Κάπως έτσι είδε το dist-upgrade που χρειάστηκε να κάνουμε σ' ένα Kali VM, το αξιολάτρευτο pfSense μας.



Load balancing δύο ιντερνετικών συνδέσεων, με ταχύτητες 10Mbps και 12,9Mbps: Κατεβάζοντας μέσω BitTorrent βλέπαμε ταχύτητες που έφταναν τα 2,8MB/sec — κι έμεναν εκεί κοντά.

COOKIES


ΛΟΙΠΟΝ, ΝΑ ΣΟΥ ΠΩ ΚΑΤΙ...
... ΕΙΧΕΣ ΔΙΚΙΟ ΤΕΛΙΚΑ ...
ΠΑΝΟΣ
... Η ΕΠΑΝΑΣΤΑΣΗ ΔΕΝ ΓΙΝΕΤΑΙ ΤΟΥΙΤΑΡΩΝΤΑΣ ΑΠΟ ΤΟΝ ΚΑΝΑΠΕ!
ΠΑΝΟΣ
... ΠΑΩ ΜΕΣΑ ΣΤΟ ΓΡΑΦΕΙΟ ΝΑ ΣΥΝΕΧΙΣΩ ΑΠ'ΤΟ ΝΤΕΣΚΤΟΠ...
... ΓΙΑΤΙ ΤΑ ΦΤΥΣΕ ΤΕΛΕΙΩΣ Η ΜΠΑΤΑΡΙΑ!

deltahacker.gr